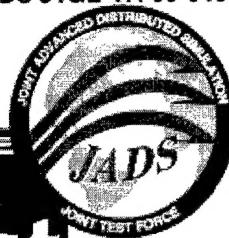


**UNCLASSIFIED**

JADS JT&E-TR-99-013

*JADS JT&E*



# Electronic Warfare Test Interim Report Phase 2

By: Maj Darrell L. Wright



September 1999

D

Distribution is unlimited.

Joint Advanced Distributed Simulation Joint Test Force . 2050A 2nd St. SE . Kirtland AFB, NM 87117-5522

DMC QUALITY INSPECTED 4

**DISTRIBUTION STATEMENT A**

Approved for Public Release  
Distribution Unlimited

A010011-3729

20000822 111

## Table of Contents

Executive Summary .....	1
1.0 Introduction .....	5
1.1 Overview.....	5
1.2 References.....	7
1.3 Electronic Warfare Test.....	7
1.3.1 EW Test Approach .....	8
1.3.2 EW Test Objectives .....	9
2.0 Phase 2 Overview.....	11
2.1 Purpose .....	11
2.2 Organizational Structure .....	11
2.2.1 Roles and Responsibilities .....	12
2.2.1.1 Deputy Director, Developmental Test and Evaluation (DD, DT&E).....	12
2.2.1.2 JADS JTF and the EW Test Team.....	12
2.2.1.3 Air Force Electronic Warfare Environment Simulator (AFEWES) 412th Test Wing .....	13
2.2.1.4 Air Combat Environment Test and Evaluation Facility (ACETEF) .....	13
2.2.1.5 Georgia Technical Research Institute (GTRI).....	14
2.2.1.6 Defense Modeling and Simulation Organization (DMSO).....	14
2.2.2 Assumptions and Constraints.....	14
2.2.2.1 Cost .....	14
2.2.2.2 Schedule .....	15
2.2.2.3 Personnel .....	15
2.3 Test Approach.....	15
2.4 Test Objectives.....	16
2.4.1 Phase 2 Test Objectives .....	16
2.5 Methodology .....	16
2.5.1 System Under Test.....	18
2.5.2 Test Scenario.....	18
2.5.3 Rules of Engagement.....	19
2.5.4 Test Configuration.....	19
2.5.4.1 Wide Area Network Components .....	20
2.5.4.2 Federate Components.....	23
2.5.5 Instrumentation .....	30
2.5.5.1 TrueTime Global Positioning System (GPS) Receiver .....	30
2.5.5.2 BanComm Timing Cards .....	30
2.5.5.3 JADS RTI Interface Logger .....	30
2.5.5.4 Network Monitoring .....	32
2.5.5.5 Network Health Check.....	33
2.5.6 Test Control and Monitoring .....	34
2.5.6.1 Test Control and Analysis Center .....	34
2.5.6.2 Site Observers.....	34
2.5.7 Runtime Infrastructure Software .....	34
2.5.8 JADS EW Test Federation Object Model .....	35
2.5.9 JADS EW Test Interface Control Document .....	35
2.6 Schedule.....	35
2.7 Security.....	35
2.7.1 Network Security .....	36
2.7.2 Data Security .....	36

3.0	Preliminary Testing Events .....	37
3.1	Phase 2 Development Tasks .....	37
3.2	Network Testing .....	37
3.2.1	Test Bed Development.....	38
3.3	RTI Performance Assessment.....	39
3.4	Phase 2 Integration.....	40
3.5	DSM Acceptance Test (DAT).....	40
3.5.1	DAT Results.....	41
3.5.1.1	Received Power Calculation Traceability.....	41
3.5.1.2	Comparison of DSM Model Results to an Independent Calculation.....	41
3.5.1.3	Timing Calibration and Traceability .....	42
3.5.1.4	Real-Time Received Power Calculation Demonstration.....	42
3.5.1.5	Real-Time Timing Calibration Demonstration .....	42
3.5.1.6	Correct ID of Threat Demonstration.....	43
3.5.1.7	Correct Response Mode Demonstration.....	43
3.5.1.8	Correct Prioritization of Simultaneous Threats.....	43
3.6	Federate Acceptance Test (FAT) .....	43
3.6.1	FAT Results .....	44
3.7	Federation Integration Test (FIT) .....	44
3.7.1	FIT Results.....	44
3.8	Verification and Validation .....	45
4.0	Test Execution .....	47
4.1	Test Control .....	47
4.1.1	Federation Time Synchronization .....	47
4.1.2	Federation Start-Up .....	48
4.1.3	Federate Status Monitoring.....	48
4.1.3.1	Digital System Model Operation .....	48
4.1.3.2	Test Control Federate (TCF) Operation.....	49
4.1.3.3	Platform Federate Operation .....	49
4.1.3.4	Radio Frequency Environment (RFENV) Federate Operation.....	50
4.1.3.5	Terminal Threat Hand-Off (TTH) Federate Operation.....	50
4.1.3.6	AFEWES Threats Federate Operation.....	51
4.2	Test Execution .....	51
4.2.1	Phase 2 Test Summaries .....	52
4.2.1.1	Notes on Day 1 .....	53
4.2.1.2	Notes on Day 3 .....	53
4.2.1.3	Notes on Day 5 .....	54
4.2.1.4	Notes on Day 7 .....	54
4.2.1.5	Notes on Day 8 .....	54
4.2.2	Federate Summaries .....	54
4.2.2.1	Digital System Model (DSM) Federate.....	55
4.2.2.2	Test Control Federate (TCF) and ADRS Operation .....	57
4.2.2.3	Platform Federate.....	57
4.2.2.4	Radio Frequency Environment (RFENV) Federate .....	58
4.2.2.5	Terminal Threat Hand-Off (TTH) Federate .....	58
4.2.2.6	AFEWES Threats Federate .....	58
4.2.3	Runtime Infrastructure (RTI).....	59
4.2.4	Wide Area Network .....	60
4.2.5	Data Collection.....	60
4.2.6	Test Control Outages .....	61

4.2.7 Test Execution Lessons Learned .....	61
4.2.7.1 Software/Hardware Reliability Issues .....	61
4.2.7.2 Test Rehearsal .....	62
4.2.7.3 DSM Performance .....	62
4.2.7.4 Site Manning/Workload During Test Execution .....	62
4.2.7.5 Tools and Procedures for Real-Time Analysis of Run Goodness .....	62
4.2.7.6 Voice Communications .....	63
4.2.7.7 Network .....	63
4.2.7.8 Test Control Procedures .....	63
4.2.7.9 Software Changes .....	64
4.2.7.10 Latency and Time Synchronization .....	64
4.2.7.11 Run Speed/Time Between Runs .....	64
4.2.7.12 RTI Heartbeat .....	64
4.2.7.13 Federate Link Health Check (LHC) .....	65
5.0 Data Analysis .....	67
5.1 ADS Measures .....	67
5.1.1 Measure 1-1-0-3. Degree to which test participants were able to distinguish between ADS (virtual or constructive) versus live (non-ADS) assets .....	69
5.1.2 Measure 1-1-0-4. Degree to which test actions were impacted due to the ability to distinguish between ADS (virtual or constructive) and live (non-ADS) assets .....	70
5.1.3 Measure 1-2-2-2. Degree to which test control procedures are impacted by ADS and how ADS can impact the pretest development and rehearsal of test control procedures .....	71
5.1.4 Measure 1-2-2-3. Degree to which data management procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data management procedures and tools .....	72
5.1.5 Measure 1-2-2-4. Degree to which data reduction and analysis procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data reduction and analysis procedures and tools .....	73
5.1.6 Measure 1-2-3-3. Degree to which ADS can increase test times, events, etc. ....	74
5.1.7 Measure 2-1-1-1. Degree to which live, virtual, and constructive entities exist, can be instrumented, and can be readied for a test .....	76
5.1.8 Measure 2-1-2-1. Average and peak throughput available for each link (JADS to AFEWES, JADS to ACETEF, and AFEWES to ACETEF) .....	78
5.1.9 Measure 2-1-2-2. Percentage of complex data types received out of order by a federate .....	79
5.1.10 Measure 2-1-2-3. Percent of total complex data types subscribed to by a federate that was received by the federate .....	80
5.1.11 Measure 2-1-2-4. Average and peak data latency .....	82
5.1.12 Measure 2-1-3-1. Degree to which test events (trials) were affected by ADS components (failure or otherwise) exclusive of network problems .....	84
5.1.13 Measure 2-1-3-2. Degree to which test events (runs) were affected by network problems (failure or otherwise) .....	86
5.1.14 Measure 2-1-3-3. Degree to which test events (trials) were affected by personnel problems .....	87
5.1.15 Measure 2-2-1-4. Ease with which data can be retrieved, post-trial, from a given node .....	89
5.1.16 Measure 2-2-2-1. Degree to which test managers can control the configurations of ADS participants, the ADS environment data, and ADS networks .....	90
5.1.17 Measure 2-3-2-3. Degree to which protocols, processes, and procedures are needed to enable effective centralized test control .....	90
5.1.18 Measure 2-3-2-4. Degree to which real-time analysis systems support test safety and other test control requirements .....	91
6.0 Correlation Analysis .....	93
6.1 EW Test Measure of Performance (MOP) Evaluation .....	93
6.2 Statistical Hypothesis Testing .....	93
6.3 Correlation Results .....	94
6.3.1 Conclusions .....	100

6.4 ADS Effects on EW Test MOP Summary .....	101
7.0 Lessons Learned.....	107
7.1 Execution Phase: Pretest.....	107
7.1.1. Software Acceptance Testing .....	107
7.1.2 Abbreviated Statements of Work (SOW) and Distributed Simulation Testing .....	108
7.1.3 Maintaining a Schedule for an Advanced Distributed Simulation Test Execution .....	108
7.1.4 Software Quality Assurance Reality .....	109
7.1.5 Strong Systems Engineering Function in ADS-Based Test Design.....	110
7.1.6 Reliable Distributor Servicing Multiple Federates.....	110
7.1.7 RTI Reliable Traffic.....	112
7.2 Execution Phase: Pretest, Test, and Post-Test .....	112
7.2.1 Conformance to a Well-Written ICD Is Necessary to Complete an ADS Exercise. ....	112
7.3 Execution Phase: Test .....	114
7.3.1 Time Synchronization .....	114
7.4 Execution Phase: Post-Test.....	115
7.4.1 RTI Best-Effort IP Multicast Groups.....	115
8.0 Conclusions/Recommendations .....	119

## Appendices

Appendix A Phase 2 Script Execution Matrix .....	121
Appendix B Site Controller Matrix .....	125
Appendix C Acronyms and Definitions .....	127

## List of Figures

Figure 1. Organizational Structure .....	12
Figure 2. Federate Nodes for Phase 2 .....	17
Figure 3. Wide Area Network Components .....	20
Figure 4. JADS EW Test Federation .....	24
Figure 5. Jammer/DSM State Transition Diagram.....	25
Figure 6. AFEWES Federate Configuration .....	26
Figure 7. EW Test Control and Analysis Center.....	28
Figure 8. HLA Logger Implementation Diagram.....	32
Figure 9. JADS 2-Node Test Bed Configuration with Communications Devices.....	38
Figure 10. JADS 3-Node Test Bed Configuration .....	40
Figure 11. Test Setup, Delays/Failures and Trial Runs .....	75
Figure 12. Aborted Trial Breakdown.....	85
Figure 13. ADS Component Problems Breakdown .....	86
Figure 14. Aborted Trial Breakdown.....	87
Figure 15. Aborted Trial Breakdown by Fault Category.....	88

## List of Tables

Table ES-1. Test Objectives .....	3
Table 1. EW Test Measures of Performance .....	8
Table 2. Phase 2 Cost Summary .....	15
Table 3. Test Objectives .....	16
Table 4. RTI Versions Tested by JADS .....	39
Table 5. Exit Criteria .....	52
Table 6. Phase 2 Test Execution Summary .....	53
Table 7. Federate software core dumps .....	55
Table 8. JADS and EW SPJ Test Objectives Correspondence Matrix .....	68
Table 9. JADS Measures Evaluated During Phase 2 .....	69
Table 10. Time Test and Run Summary .....	75
Table 11. Scheduled Test Time Compare to Number of Test Events Completed .....	76
Table 12. Average and Peak Packet Rate and Load Values .....	78
Table 13. Lost Data Traffic Messages by Link .....	81
Table 14. Node-to-Node Traffic Latency by Data Element (milliseconds) .....	83
Table 15. Impact of ADS Component Problems .....	85
Table 16. Impact of ADS Network Problems on Trial Events .....	87
Table 17. Impact of ADS Component Problems on Trial Events .....	88
Table 18. Correct ID Response Time Correlation Matrix .....	95
Table 19. Correct ECM Technique Selection Response Time Correlation Matrix .....	96
Table 20. RMS Tracking Error Correlation Matrix .....	97
Table 21. Jamming-to-Signal Ratio Correlation Matrix .....	97
Table 22. Number of Breaklocks Correlation Matrix .....	98
Table 23. Reduction in Engagement Time Correlation Matrix .....	99
Table 24. Reduction in Missiles Launched Correlation Matrix .....	99
Table 25. Missile Miss Distance Correlation Matrix .....	100
Table 26. Effects of ADS on EW Test Measures of Performance .....	101



## EXECUTIVE SUMMARY

### **1.0 Introduction**

This summary serves as a stand-alone document, as well as part of this report. Therefore, there is some duplication of text as well as tables and figures between this summary and the full report.

### **2.0 JADS Overview**

The Joint Advanced Distributed Simulation (JADS) Joint Test and Evaluation (JT&E) was chartered by the Deputy Director, Test, Systems Engineering and Evaluation (Test and Evaluation)<sup>1</sup>, Office of the Secretary of Defense (Acquisition and Technology) in October 1994 to investigate the utility of advanced distributed simulation (ADS) technologies for support of developmental test and evaluation (DT&E) and operational test and evaluation (OT&E). The JADS Joint Test Force (JTF) is Air Force led with Army and Navy participation. The JADS JT&E program is scheduled to end in March 2000.

The JADS JTF investigated ADS applications in three slices of the test and evaluation (T&E) spectrum: ADS support of air-to-air missile testing; ADS support for command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) testing; and the Electronic Warfare (EW) Test which explored ADS support for EW testing.

### **3.0 EW Test Overview**

The tasking to conduct an ADS-based EW test called for an airborne self-protection jammer (SPJ) as the system under test (SUT). The emphasis of the EW Test was on the performance of the ADS components and their contribution or impact to testing rather than on the performance of the SPJ pod itself. Measures of performance (MOPs) for the SPJ were identified as measures that would most likely be affected by distributed testing. Statistical comparison of the MOPs became the methodology for evaluating ADS. JADS evaluated distributed test control and analysis, network performance, relationships between data latencies, and ADS-induced data anomalies. Time, cost, and complexity, as well as validity and credibility of the data, were part of the evaluation.

The EW Test was designed as a three-phase effort. The first phase provided a baseline of jammer performance data in a non-ADS environment that was then compared to the data collected in the second and third phases using an ADS environment. The second phase used a digital system model of the SPJ representing an early developmental test. The third phase used the SPJ mounted on the aircraft that was suspended in an installed systems test facility. This test represented a combined integration and effectiveness test that would occur late in the SPJ development.

---

<sup>1</sup> This office is now the Deputy Director, Developmental Test and Evaluation (DD, DT&E).

Phase 1 included a risk reduction flight test effort at the Western Test Range (WTR) to define a reference test condition; 14.4 hours of baseline flight test using a modified ALQ-131 jamming pod at the WTR; a nine-day hardware-in-the-loop (HITL) test at the Air Force Electronic Warfare Environment Simulator (AFEWES) at Fort Worth, Texas; and a three-day system integration laboratory (SIL) test at the Automatic Multiple Environment Simulator (AMES) facility at Eglin Air Force Base (AFB), Florida. The HITL and SIL tests were added to supplement the baseline flight testing and to provide missing data. This established the baseline of environment and jammer performance data against two command-guided surface-to-air missile (SAM) sites, one semiactive surface-to-air missile site, and one anti-aircraft artillery (AAA) site. The reference test condition and baseline data were used to develop the ADS test environment for the two subsequent ADS test phases and provided the baseline data for comparison with the ADS test results.

Phase 2 (the subject of this report) was a test of a real-time digital system model (DSM) of the modified ALQ-131 receiver processor linked with terminal threats at the AFEWES facility and a scripted model of the terminal threat hand-off portion of an integrated air defense system (IADS). The reference test condition used in the Phase 1 flights was replicated as closely as possible in the synthetic ADS environment; the jammer model was flown, via the scripted flight profiles developed from the actual open air range (OAR) baseline flights and HITL test, against the AFEWES threats positioned in the synthetic environment as the threats were positioned on the range.

Phase 3 will be a test using the modified ALQ-131 jammer installed on an F-16 aircraft in the Air Combat Environment Test and Evaluation Facility (ACETEF) located at Patuxent River Naval Air Station, Maryland. This facility will be linked with AFEWES threats using the same reference test condition as the previous test and controlled by the same scripted flight profile.

## **4.0 Overview of EW Test Phase 2**

### **4.1 Purpose**

The primary purpose of Phase 2 was to collect SPJ performance data using a DSM representation of the jammer in an ADS-based test environment. The performance data were combined with data collected on the ADS environment itself to determine how much of an impact ADS had on the test. Phase 2 test objectives are summarized in Table ES-1.

**Table ES-1. Test Objectives**

<b>Obj #</b>	<b>Objective</b>
1-1	Establish performance in JADS/DSM environment
1-2	Establish the repeatability of DSM test results
1-3	Establish ranges of DSM statistics for event data
1-4	Establish range of correlation coefficients for series observables
1-5	Quantify the effects of data latency on JADS/DSM test environment
1-6	Quantify the operating reliability and mean time between failure of the JADS network
1-7	Determine the connectivity performance of the JADS network

## **4.2 Approach**

The overall test approach was designed to provide a means of capturing the ADS effects within the Phase 2 architecture. The high level architecture (HLA) was used to link the DSM located at ACETEF, HITL terminal threats at the AFEWES facility, and other models hosted in the JADS test control facility. The test collected data for subsequent comparison with the EW Test MOPs collected in Phases 1 and 3 as well as the ADS data needed to calculate ADS MOPs. A statistical comparison was used to compare the EW Test MOP data sets. The results of the specific EW Test MOPs are classified and are reported in a separate document. The statistical comparisons of the MOPs resulted in a correlation measure called a “P-value.” P-values are unclassified and are included in this report.

## **5.0 Phase 2 Test Results**

### **5.1 Fulfillment of Test Objectives**

All Phase 2 test objectives except 1-5 were met. JADS was unable to quantify the effects of data delay on the JADS/DSM environment beyond observing that the effects were too small to be measured. Other sources of variance overwhelmed the effects of data delay.

### **5.2 General Results**

Phase 2 used an HLA-compliant ADS architecture to successfully recreate both an open air test and hardware-in-the-loop test. The architecture successfully integrated a DSM representing an early representation of a self-protection jammer with the high fidelity threats at AFEWES. This implies that ADS may be used to address the EW test process limitations. A complete discussion on the utility of ADS to EW testing will be the subject of the JADS EW Test final report.

Test results and operator interviews indicated that even though there were some isolated incidents of ADS impacting results, there were no consistent ADS-induced biases or flaws that made the data invalid. Data latency in excess of the design goal and lengthy bursts of lost aircraft position

data did not affect the EW Test MOPs in any consistent, measurable fashion. Subject matter experts confirmed that the data produced by the JADS architecture were valid. This implies that properly designed ADS architectures will not impact test results.

There were limitations within the ADS architecture that JADS created. Different jammer techniques and more reactive players required that the bursts of lost aircraft position data be resolved and latency performance be improved over what was observed in Phase 2. Predictive jammer techniques would also require more of the jammer processing logic to be collocated at AFEWES. Several of the message structures and common data used in our test would have to be examined before being used in other tests. While all the message structures have room for growth, they need to be examined by future implementers to ensure the size and intent meet the requirements of the new federation.

The most significant limitation to this architecture was the availability of threats suitable for ADS-based testing. Low fidelity threats are not difficult to add to this architecture, but they have to run in real time to interact properly with the manned threats. Models are not sufficient to address shortfalls of the EW test process since they do not recreate the largest source of variation - human operator actions. Manned high fidelity threat representations are the key to obtaining the highest benefit from this architecture. The AFEWES facility uses distributed simulation techniques within its facility to accomplish traditional testing. ADS simply allows AFEWES to connect to other facilities or locations. The OAR used in Phase 1 had high fidelity threat simulators as well. However, these were not suitable in their current configuration to accomplish testing within the JADS architecture. Radio frequency injection into the threat for both target and jamming is key to making these threat assets available using ADS.

## 6.0 Conclusions

Phase 2 demonstrated that ADS tests create valid EW test data when properly designed. ADS can be used to connect real-time digital system models with manned threat simulators. This makes ADS a potentially feasible tool for EW testers. However, the availability of suitable manned simulators will likely determine how quickly ADS is integrated into the mainstream of EW testing.

## **1.0 Introduction**

### **1.1 Overview**

The Joint Advanced Distributed Simulation (JADS) Joint Test and Evaluation (JT&E) program is an Office of the Secretary of Defense (OSD)-sponsored joint service effort designed to determine how well advanced distributed simulation (ADS) can support test and evaluation (T&E) activities. The Electronic Warfare (EW) Test is one of three tests comprising the JADS Joint Test Force (JTF). It was chartered separately in 1996 to test the utility of geographically distributed simulations to the EW T&E community. This report focuses on results of the EW system Phase 2 testing using the high level architecture (HLA).

The JADS EW Test was designed to provide insight into ADS-based testing. JADS was charted to address three issues:

- What is the present utility of ADS for T&E?
- What are the critical constraints, concerns, and methodologies when using ADS for T&E?
- What are the requirements that must be introduced into ADS systems if they are to support a more complete T&E capability in the future?

These issues were mapped to activities within the EW Test effort. This mapping first appeared in the 1996 Analysis Plan for Assessment. Further refinement is described in the 1998 Program Level Test Activity Plan/Data Management and Analysis Plan (TAP/DMAP). Execution of the test brought further refinements that are summarized in Tables 8 and 9.

The JADS EW Test methodology fully incorporated Department of Defense's (DoD) high level architecture, which requires some description as to how it relates to the Phase 2 methodology. HLA is an object-oriented approach to developing interactive simulation models and environments. The HLA consists of design and execution tools, interfaces, and design rules to facilitate interfacing simulation applications. HLA is intended to provide a common framework within which a specific architecture can be implemented. For each simulation, an object model is built providing an appropriate abstraction of the objects, attributes, associations, and interactions used by the simulation. JADS EW Test used multiple interacting simulations to form what is called an HLA federation (for more information about HLA, see the HLA web site <http://hla.dmso.mil/>). This set of interacting simulations or federates, along with their respective object models, is described in a federation object model or FOM. The JADS EW Test federation was used with a supporting HLA runtime infrastructure (RTI) to execute an ADS-based test representing the EW open air range (OAR) Phase 1 test and range environment.

ADS was expected to bring specific benefits to the EW test process, a formally documented, systematic test process covering all phases of system development. It served as the template for the simulation, test and evaluation process (STEP) adopted by OSD. The JADS feasibility study identified three shortfalls in implementing the EW test process that ADS might solve.

The first shortfall is the inability to correlate test results throughout the development process. The EW test process recommends a model, test, model approach. Limitations in both facilities and models result in fidelity differences in both the system under test (SUT) and the threats because of their continuing evolution. Too many variables change from test event to test event to trace apparent performance changes that are due to threat differences between facilities or SUT design evolution. ADS holds the promise of allowing a fixed set of high fidelity threats to be used throughout the development process. Limiting the threat representation to one set of high fidelity threats implies all performance differences would be due to SUT evolution. This in turn would allow statistical comparisons to aid decision makers in better understanding system performance.

The second shortfall relates to correlation as well: test resource fidelity. Testing against lower fidelity threats may allow SUT problems to go undetected until later stages of testing. However, high fidelity test resources such as man-in-the-loop threat simulators are expensive and are available at very few facilities. Therefore, there is very little duplication of the highest fidelity resources. The tester is often forced to use low fidelity threat simulators or models early in system development. Testing against high fidelity threats requires the system to be transported to the appropriate facility and integrated. Transportation can be impractical with bread-board and brass-board hardware. Testing against models precludes operator interaction effects with the SUT. For jammers, this interaction is critical. ADS holds the promise of allowing the SUT to interact with the high fidelity resource without collocating them. This would allow early representations of the SUT to interact with high fidelity threat resources. Any real-time representation of the SUT, including digital system models, could be used for testing. This would allow system designers to see critical interactions, such as operator actions, very early in the design process.

The third shortfall is availability of resources. Test facilities are limited by budget realities that force them to limit testing to specific capabilities. There is no single test facility that provides the tester with all the high fidelity resources and other support needed to completely test complicated EW systems. This is especially true of jammer systems. ADS holds the promise of allowing the tester to link together the resources needed to accomplish the test no matter where the resource is located. This would allow traditionally separate serial tests to be conducted in coordination with one another.

The EW Test was designed around three test phases to address both the JADS issues and the ability of ADS to solve the three EW test process shortfalls discussed above. Phase 1 used traditional test methods and environments to establish a performance baseline of an operational airborne self-protection jammer against four threats. This phase was accomplished in three different environments. These separate environments were needed to overcome test instrumentation limitations and procedure problems that prevented JADS from measuring all the performance measures in a single environment. Jammer effectiveness measures were collected in both the OAR and the Air Force Electronic Warfare Evaluation Simulator (AFEWES) facilities. Jammer internal response times were measured in a system integration lab. These results are reported in classified and unclassified reports.

Phase 2 used a digital system model (DSM) to represent the jammer. The DSM was hosted at the Air Combat Environment Test and Evaluation Facility (ACETEF), Patuxent River, Maryland, and geographically separated from the threats at AFEWES, but it had to interact with the threats to recreate the baseline data. Great care was taken to ensure the same reference test condition was used. Several of the key components of Phase 2 were scripts derived from actual OAR recorded data or developed from the reference test condition flown on the range. Statistical correlation of the EW Test measures of performance (MOP) were used to compare the ADS results with the traditional test results obtained in Phase 1. The results of the statistical comparison were expected to provide insight into how much ADS impacted the test results.

Phase 3 will use the same components as Phase 2 except for the DSM, which will be replaced with the real jammer installed in the ACETEF facility. The real jammer required JADS to use the HLA interface to allow radio frequency (RF) stimulators to recreate the action of the RF environment for the jammer. The same EW Test measures of performance will be collected as have been in the previous phases. Statistical comparison of the EW Test MOPs will again be used to compare the ADS results with one another and with the traditional test results. The comparison is expected to provide insight into how well the ADS results could be repeated and how much ADS impacts the test results. Phase 3 results will be presented in a later report.

Phase 2 was not expected to provide complete answers to the issues that must be addressed by the EW Test. This is an interim report limited to test execution, unclassified ADS measure results, unclassified correlation results, and lessons learned. The complete answer to the JADS issues and to the ability of ADS to address the EW test shortfalls will be presented in the Phase 3 report. A complete presentation of the EW Test MOP results will be made in a separate classified, combined Phase 2/Phase 3 report. This report will also address the EW Test shortfalls to provide the EW community with a single reference source.

Additional background information on the history and planning for the EW Test in general and the Phase 2 effort specifically is contained in the references listed below.

## **1.2 References**

Electronic Warfare Test Analysis Plan for Assessment (APA), May 1996.

Program Level Test Activity Plan and Data Management and Analysis Plan (TAP/DMAP), March 1998.

Electronic Warfare Phase 2 TAP/DMAP, November 1998.

## **1.3 Electronic Warfare Test**

The tasking to conduct an ADS-based test of an EW system specifically called for the use of an airborne self-protection jammer (SPJ) as the surrogate system under test (SUT). The actual SUT for JADS is ADS. In the summer of 1995, JADS presented a comprehensive test and analysis approach for an EW Test to the technical advisory board (TAB) and the senior advisory council

(SAC). The JADS EW Test approach was fully supported by the TAB but not chartered initially primarily because of the high cost (\$18 million). In response, JADS tailored the initial EW Test design and subsequently developed a reduced scope, lower cost test and analysis approach using a modified ALQ-131 SPJ pod as the surrogate SUT. For this test, the ALQ-131 was modified to operate with tailored preflight and mission software tapes that affected its operational performance. Jamming effectiveness results during the JADS EW Test may not be representative of operational pods in the tactical inventory.

The emphasis of the EW Test was on the performance of the ADS components and their contribution to testing rather than on the performance of the modified ALQ-131 test item itself. MOPs for the jammer were computed only as a means of testing ADS. These measures are listed in Table 1. JADS evaluated distributed test control and analysis, network performance, relationships between data latencies, and ADS-induced data anomalies. Time, cost, and complexity, as well as validity and credibility of the data, were part of the evaluation. Specific test scenarios were selected to allow this comparison. Additionally, some test activities were planned that would not be feasible without ADS technology.

**Table 1. EW Test Measures of Performance**

MOP #	Description
1	Correct threat identification
2	Correct threat identification response time
3	Correct electronic countermeasures (ECM) technique selection
4	Correct ECM technique selection response time
5	Jamming-to-signal ratio
6	Root mean square (RMS) tracking error
7	Number of breaklocks
8	Reduction in engagement time
9	Reduction in missiles launched
10	Missile miss distance

**NOTE:** The integrated product team redefined MOP 6 to better reflect SUT performance.

### 1.3.1 EW Test Approach

The EW Test was designed as a three-phase effort providing a baseline of SUT performance data in a non-ADS environment that was then compared to multiple tests of the same configuration in an ADS environment. The HLA was used in Phases 2 and 3.

Phase 1 included an OAR risk reduction flight test effort; baseline flight test using a modified ALQ-131 jamming pod at the Western Test Range (WTR); a short hardware-in-the-loop (HITL) test at AFEWES at Fort Worth, Texas; and a system integration laboratory (SIL) test at the Automatic Multiple Environment Simulator (AMES), Air Warfare Center (AWC), Eglin Air

Force Base (AFB), Florida. The purpose of this test phase was to establish a baseline of environment and SUT performance data against two command-guided surface-to-air missile (SAM) sites, one semiactive surface-to-air missile site, and one anti-aircraft artillery (AAA) site. This scenario was used to develop the ADS test environment for the following phases and provided the baseline data for comparison with the ADS test results. Additionally, the performance data provided a baseline for attempting to compare the data across all three phases of the test. The Phase 1 test scenario was very structured and constrained to provide the greatest opportunity for repeatability and, therefore, the greatest opportunity for identifying ADS effects through statistical comparison.

The ADS-based tests, Phases 2 and 3, used a real-time DSM representing the SUT (Phase 2) and will use the installed modified ALQ-131 on an F-16 (Phase 3) located in an installed systems test facility (ISTF), respectively. The simulated threat environment and engagements closely resembled the OAR test for Phase 2. The same threat environment and engagements will be used during Phase 3. The baseline data collected in Phase 1 were used to create the synthetic replication of the aircraft as well as the engagement conditions.

### **1.3.2 EW Test Objectives**

It is difficult to measure ADS utility in the real world of EW T&E. There are significant technical challenges in implementing ADS in this environment as well as programmatic issues such as cost and schedule impacts. The achievable (not just theoretical) performance that can be obtained by inserting ADS into the established EW test process must be determined. The overall objective of the JADS EW Test is to address these questions and thus assess the utility of ADS to EW test and evaluation. Specific test objectives are listed in the JADS EW APA and Program Level TAP/DMAP (Section 2.4.1). Phase 2 was an interim effort, and this report will not address all the objectives listed in the APA or the TAP/DMAP.



## **2.0 Phase 2 Overview**

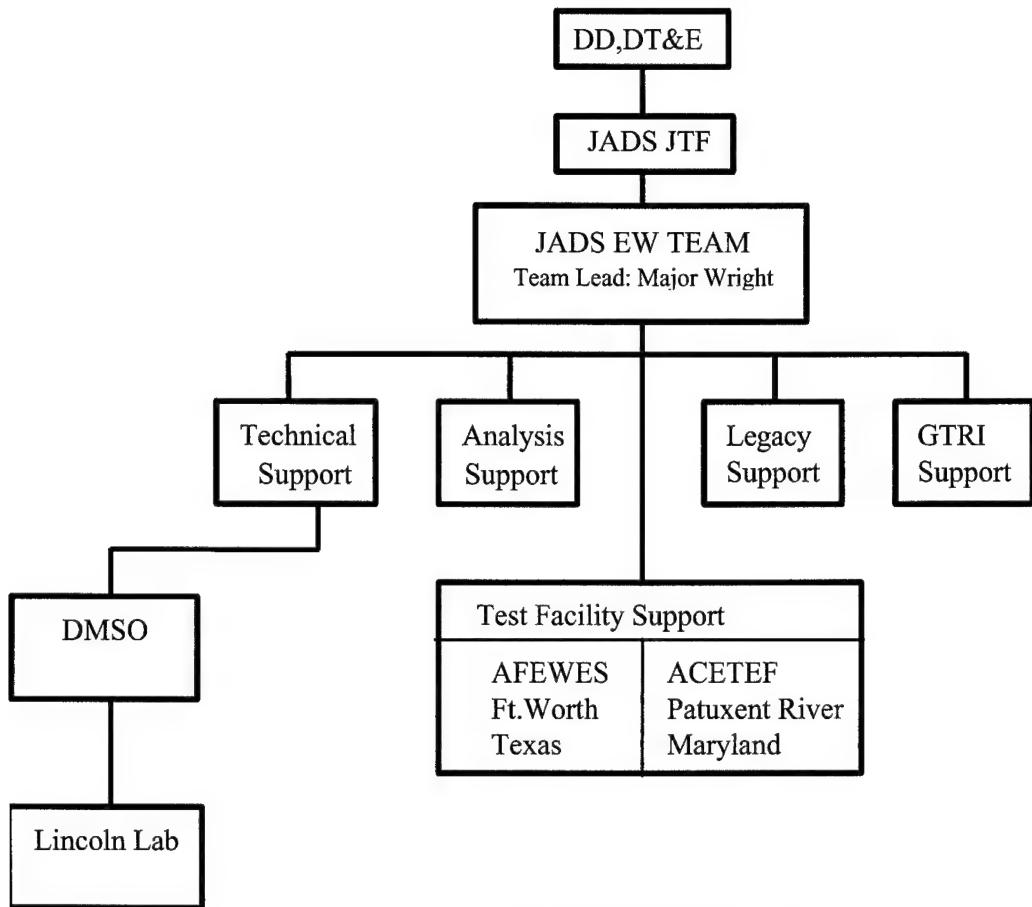
### **2.1 Purpose**

Phase 2 collected SPJ performance data using a DSM representation of the jammer in an ADS-based test environment. The HLA was used to link the DSM located at ACETEF, HITL terminal threats at the AFEWES facility, and other models hosted in the JADS test control facility. The test collected data for subsequent comparison with MOP data collected in Phases 1 and 3. Specifically, the analysis of Phase 2 results included descriptive statistics on the DSM and ADS MOPs. A correlation analysis approach was used to compare the MOP data sets. The overall approach was to provide a means of capturing the ADS effects within the Phase 2 architecture. The results of the specific EW Test MOPs are classified and will be reported separately. Correlation of the MOPs results in a correlation measure called a “P-value” which is not classified. Those results are part of this report.

### **2.2 Organizational Structure**

The JADS EW Test was planned and executed using an integrated product team (IPT). This arrangement allowed the test participants to have a strong voice in how the test would be executed and reported while allowing JADS to best use the electronic warfare expertise of each organization. The IPT was comprised of the EW Test team lead, the Georgia Technical Research Institute (GTRI) team lead, the AFEWES team lead, the ACETEF team lead, the 413<sup>th</sup> Flight Test Squadron, Edwards AFB, California, team lead, and the Air National Guard Air Force Reserve Test Center (AATC) team lead.

Figure 1 shows the IPT depicted within the JADS conventional organizational structure for coordinating and reporting during Phase 2 of the EW Test.



**Figure 1. Organizational Structure**

## 2.2.1 Roles and Responsibilities

### 2.2.1.1 Deputy Director, Developmental Test and Evaluation (DD, DT&E)

- Oversaw the JADS JT&E
- Approved the program test plan (PTP)
- Approved JADS financial requirements
- Oversaw the analysis and reporting of test results

### 2.2.1.2 JADS JTF and the EW Test Team

- Developed the federation execution planners workbook (FEPW)
- Developed the federation objective model (FOM)
- Developed the data logger software
- Managed the interface control document (ICD)
- Developed ADS measures and identified related data elements

- Acquired, installed, and supported communications routers, hubs, and switches
- Implemented and conducted benchmarks of the RTI, computer, and communications architecture to support JADS latency requirements
- Managed funding to accomplish the test
- Acquired, verified, and supported usage of T-1 long haul communications circuits
- Developed software tools for analyzing data and processing logger files
- Installed and integrated computer capabilities in the Test Control and Analysis Center (TCAC) and other sites
- Developed and integrated the components of the EW Test environment
- Developed and provided AFEWES, ACETEF, and GTRI with the Phase 2 TAP/DMAP
- Coordinated, rehearsed, and controlled execution of Phase 2 test activities
- Operated the TCAC during tests
- Analyzed and evaluated Phase 2 data and MOP
- Reported interim and final results to OSD

#### ***2.2.1.3 Air Force Electronic Warfare Environment Simulator (AFEWES) 412th Test Wing***

- Developed the AFEWES federate software and integrated with the RTI, federate logger software and other JADS federation components
- Provided aircraft radar cross-section (RCS) data to GTRI for DSM development
- Provided Phase 2 test facilities and test management personnel
- Provided use of the Tactical Air Mission Simulator (TAMS)
- Provided simulated terminal threats and system operators
- Provided the JammEr Techniques Simulator (JETS) for radio frequency signals to the threats
- Provided threat test management centers (TMC) for data collection
- Provided data, videotapes, and strip charts for each simulator
- Provided threat and SUT antenna patterns to GTRI
- Provided subject matter expert (SME) for verification and validation (V&V) of the distributed environment

#### ***2.2.1.4 Air Combat Environment Test and Evaluation Facility (ACETEF)***

- Provided Phase 2 integration software support
- Provided Phase 2 test facilities for the communications and DSM hardware and software
- Provided Phase 2 network personnel support
- Provided additional Phase 2 DSM operators

#### **2.2.1.5 Georgia Technical Research Institute (GTRI)**

- Developed the ICD
- Developed the requirements/specifications for federates and DSM (platform, terminal threat hand-off, test control, DSM, and RF environment)
- Developed federates and DSM
- Developed script generation tools
- Modified the Automated Data Reduction Software (ADRS) tool
- Developed and calibrated the DSM software
- Provided support for federation implementation and integration at JADS
- Provided test execution support at AFEWES
- Developed test control and execution methodology and automated capabilities
- Provided analysis and technical support for test data reduction and correlation
- Provided SME for V&V of the distributed environment

#### **2.2.1.6 Defense Modeling and Simulation Organization (DMSO)**

- Provided RTI technical support
- Provided access to the RTI developers
- Delivered an RTI that met JADS latency requirements

### **2.2.2 Assumptions and Constraints**

During the test design phase, JADS developed the EW Test by applying a set of goals and constraints relative to test content, cost, schedule, and personnel described in the EW APA. The primary programmatic constraints were cost and schedule. In some ways, this test was a simple example of cost as an independent variable. Technical content of the test was the primary area available for trade to maintain cost and schedule. Technical limitations (imposed because of cost and schedule constraints) resulted in a limited set of open air range instrumentation and constrained rules of engagement (ROE) being used to support a single reference test condition (RTC) during the Phase 1 baseline data collection. Each of these technical constraints is discussed in the content constraints section of the Phase 1 TAP/DMAP. Phase 2 duplicated the same RTC as used in Phase 1. The impacts of the nontechnical constraints, cost, schedule, and personnel, are discussed in separate sections. The net result of the constraints is that the RTC and threat systems engagement represent a simple subset of developmental testing for an EW SPJ. This subset is sufficient for examining the impact of ADS on EW testing.

#### **2.2.2.1 Cost**

JADS had an established test budget and designed Phase 2 within the budget starting from original cost estimates. Because of funding limitations established for the EW Test, the resulting design represented the minimum DSM test required to evaluate the utility of applying ADS to EW T&E. Although two straight weeks of testing was not the best approach (this allowed very little

time to assess or correct anomalies), based on our allotted time frame coupled with AFEWES facility availability and the budget, this was our best, most affordable option.

**Table 2. Phase 2 Cost Summary**

Cost Item	Amount
AFEWES support	\$1,205,000
GTRI	\$ 970,000
ACETEF	\$ 337,000
Phase 2 network and other JADS costs	\$ 285,000
Phase 2 total	\$2,797,000

#### **2.2.2.2 Schedule**

The primary schedule constraint associated with the Phase 2 effort was related to the schedule of the overall JADS EW Test program. The current JADS JTF charter has personnel assigned through the end of FY99. The EW Test was designed for completion within the current charter. Phase 2 had to be completed in time (not later than December 1998) for the ISTF test (Phase 3) and all reports to be completed before 1 October 1999. It was imperative that all major events required for Phase 2 occur within a very limited planning window. Phase 2 was originally scheduled three weeks earlier than executed. Delays in Phase 1 data collection forced the schedule to slip from November 1998 to December 1998. Phase 1 consumed several months, not in succession, collecting the SPJ timing data used to calibrate the DSM. Three separate test facilities and designs were used before obtaining accurate, useable data. Additional delays in the program would have severely hampered our ability to successfully conduct Phases 2 and 3.

#### **2.2.2.3 Personnel**

The final constraint on Phase 2 was to conduct the test program using current JTF assigned personnel augmented by experienced contractor and test facility personnel. Adequate personnel were available to complete Phase 2. However, the plan was designed to have only two active threats at AFEWES because of personnel limitations—four manned threats were active for approximately three hours during the entire test period. The test team compensated for this shortfall by using the federate originally intended to control unintended emissions measured in Phase 1 to publish the emissions from the unmanned threats. This limitation did not adversely affect the DSM operating environment. On-site support during integration and test execution was reduced from the original planning level to contain cost.

### **2.3 Test Approach**

Phase 2 tested a real-time digital system model of the modified ALQ-131 receiver-processor linked with HITL terminal threats at the AFEWES facility and a scripted model of the terminal threat hand-off portion of an integrated air defense system (IADS). The threat laydown from the

OAR was replicated in the synthetic ADS-based environment and the SUT was flown, via the scripted flight profiles developed from the actual OAR flights and initial HITL test, against the AFEWES threats. This phase evaluated the ability to apply increased fidelity and resources through ADS early in the development cycle of a SPJ system and to develop and refine requirements for a new system through actual effectiveness testing of a proposed system in digital model form.

## 2.4 Test Objectives

The EW Test objectives were used to assess the utility of ADS. There were significant technical challenges in implementing ADS in this environment as well as programmatic issues such as cost and schedule impacts. It is difficult at best to derive the utility assessment within the EW T&E framework without measurable objectives that provide insights for cost savings or value added. The overall test objectives are outlined in the JADS EW APA and Program Level TAP/DMAP. Excerpts of the test objectives that apply directly to Phase 2 are listed in Table 3.

### 2.4.1 Phase 2 Test Objectives

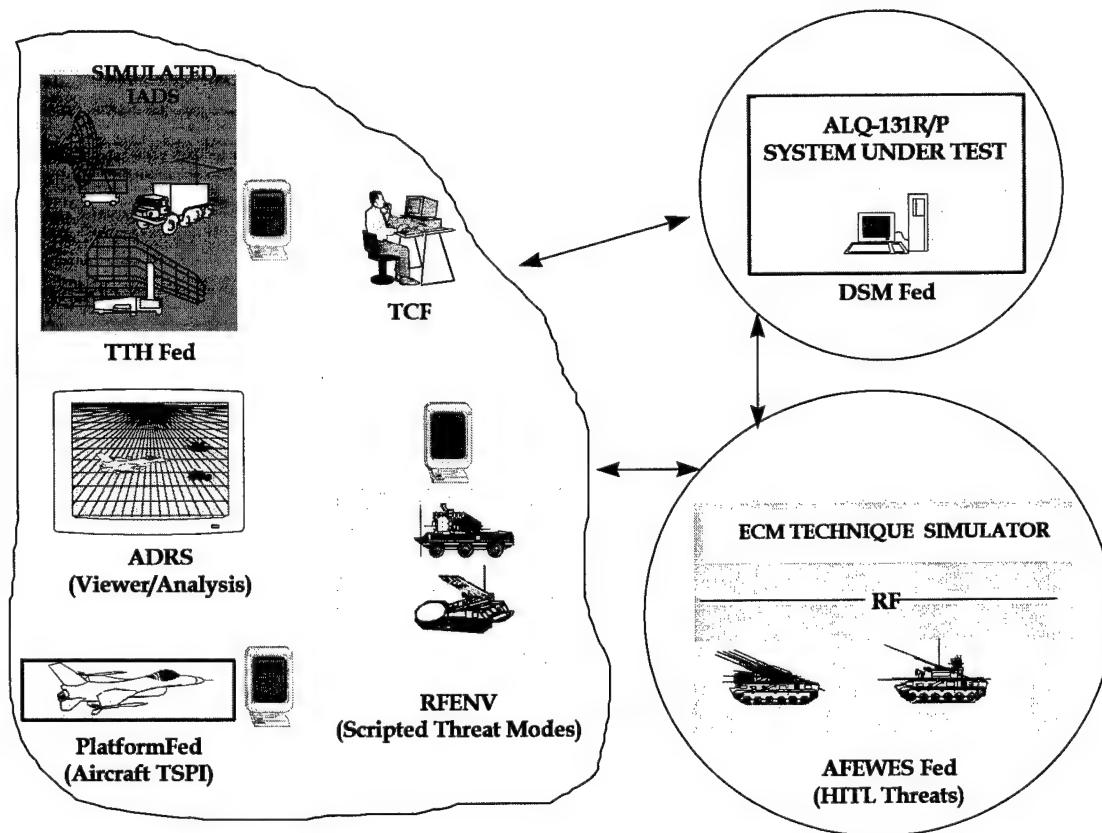
**Table 3. Test Objectives**

Objective #	Objective
1-1	Establish performance in JADS/DSM environment
1-2	Establish the repeatability of DSM test results
1-3	Establish ranges of DSM statistics for event data
1-4	Establish range of correlation coefficients for series observables
1-5	Quantify the effects of data latency on JADS/DSM test environment
1-6	Quantify the operating reliability and mean time between failure of the JADS network
1-7	Determine the connectivity performance of the JADS network

## 2.5 Methodology

Phase 2 used seven federates consisting of one model and the AFEWES threat simulators to simulate the engagements. These seven federates were platform, terminal threat hand-off (TTH), digital system model, test control, radio frequency environment (RFENV), analysis, and threats (commonly referred to as the AFEWES federate). The federates and simulators were linked to form a federation. The platform federate published the F-16 aircraft time-space-position information (TSPI) data collected from the Phase 1 OAR flight profiles. The AFEWES threat simulators, representing the Simulated Air Defense System (SADS) III, SADS VI M, SADS VIII, and the Weapon Evaluated Simulated Threat (WEST) X simulated threats used in Phase 1, were controlled via scripted commands sent by TTH federate and subsequently cued by voice to human operators. All four threat simulators disengaged the aircraft at each end of the flight profile replicating the OAR test. A DSM of the modified ALQ-131 Block II receiver/processor (R/P) was used to represent the SUT. Figure 2 illustrates the federate representations (nodes) for Phase

2. To maintain the integrity of the DSM responses, the modes of the scripted (nonlive) simulated threats were sent by the RFENV federate, thereby having the DSM manage four threats. This test was observed through the use of two software tools. The analysis federate visually depicted the engagement while displaying real-time threat mode changes and limited MOP data. ADRS, which is part of the test control federate (TCF), also displayed threat mode data along with critical threat parameters.



**Figure 2. Federate Nodes for Phase 2**

Two types of federates were used for Phase 2: playback federates and gateway federates. They are differentiated by their computer architecture and operational function. Gateway federates were responsible for connecting non-HLA compliant facilities and applications to the JADS federation. The federate that passed data to and from the threats at AFEWES is one example of a gateway. Other examples used the same code as the playback federates to interface non-HLA software with the federation. ADRS was an off-the-shelf product that JADS had modified for the test and the jammer model reused 85% of its code from existing software. JADS decided that neither needed to be HLA compliant. GTRI elected to create a single application that acted as both a gateway and to playback scripts. JADS called these playback and pass-through federates. The two pass-through federates used in Phase 2 were the jammer/DSM and TCF federates. The DSM federate was responsible for passing data to and from the software simulation of the jammer. The TCF

was responsible for starting the federation execution of a trial under control of the test director and displaying relevant test data for monitoring the test execution and evaluating test measures of performance. The TCF had an Silicon Graphics, Inc., (SGI) O2 computer hosting the UNIX-based RTI interface software linked to multiple personal computers (PC) running Automated Data Reduction Software (ADRS) application software written in C++ language under Windows 95. The pass-through federate software on the SGI O2 was responsible for ensuring that the ADRS was supplied with all the necessary data to perform data reduction, analysis, and test visualization.

Playback federates are designed to model an important OAR component by playing back a data script of key interactions or attributes recorded in Phase 1 test events - hence their designation. Playback federates in Phase 2 were the platform, RF environment, and terminal threat hand-off. Playback federates were responsible for publishing data elements from predefined scripts of data attributes and interactions to the JADS federation at correct times during the simulation. The playback data were loaded during the joining process and transmitted based on a timed sequence of events. For each platform script generated there was a corresponding TTH and RFENV script for the normal runs. However, during the ADS excursion runs, all four simulated threats were manned at AFEWES; there were no scripts generated for or published by the RFENV federate.

### **2.5.1 System Under Test**

The modified ALQ-131 SPJ representation used for Phase 2 was a digital system model developed by the GTRI. This DSM was not HLA compliant. GTRI elected to create a gateway (pass-through) software application. The DSM federate was comprised of both the model and the gateway software—each hosted on separate computers. This federate simulated the critical functions of the modified ALQ-131 R/P in the JADS EW Test HLA federation. The DSM used a combination of databases and measured ALQ-131 internal timing distributions to represent ALQ-131 behavior consistent with that observed in Phase 1. The DSM model was developed using algorithms from the general effectiveness model (GEM) and the Joint Modeling and Simulation System (JMASS) receiver programs. The DSM model resided on a Pentium PC processor with Ethernet and universal time code (UTC) interface capabilities, and the pass-through software resided on an SGI 02 workstation. Both computers were linked by Ethernet and were located at the ACETEF node for Phase 2.

### **2.5.2 Test Scenario**

The F-16 aircraft was represented in the test by an RCS look-up table used inside AFEWES and a script that contained actual TSPI recorded in Phase 1. The script was used by an instance of the playback application called the platform federate. Each script replicated flying one simple profile at 360 knots at an altitude of 9,000 feet mean sea level (msl) identical to the Phase 1 OAR flight profile. The AFEWES threat simulators, representing the SADS III, SADS VI M, SADS VIII, and the WEST X comparable to those used in Phase 1, were activated and deactivated via scripted voice commands. Human operators in each of the manned simulators responded to the voice commands. AFEWES operated the live simulated threats in pairs—SADS VIII/WEST X and SADS III/SADS VI M. These particular pairs were chosen to accommodate AFEWES

manning considerations. In order to provide a controlled experiment to evaluate utility of ADS, a set of ROE was used during all phases of the JADS EW Test. These rules were intended to constrain the WTR operator actions to those that could be accomplished at the AFEWES facility while allowing some freedom to engage the aircraft. Additionally, because of limited operators at AFEWES, all four threats could not operate simultaneously. Therefore, RFENV transmitted the scripted modes of the inactive threat pair (SADS III and SADS VI M) coincident with the site controller matrix (see Appendix B), while AFEWES operated the other pair (SADS VIII and West X). Once a sufficient amount of data was collected the previous non-HITL threats became active at AFEWES (SADS III and SADS VI M), and the RFENV federate generated the modes for the initially active threats now inactive (SADS VIII and West X). The threat simulators engaged and disengaged the aircraft on each northbound and southbound flight profile which replicated the OAR test. Before execution began with the second pair of active threats, AFEWES conducted several engagements with all four threats operational within their facility. These runs were identified as ADS excursions. The ADS excursions were only used for ADS analysis and not for SPJ MOP calculations.

### **2.5.3 Rules of Engagement**

The Phase 2 ROE were driven by the requirement to rerun the OAR engagements using the Phase 2 architecture. The description of the ROE used for Phase 1 is delineated in Appendix A of the Phase 1 TAP/DMAP.

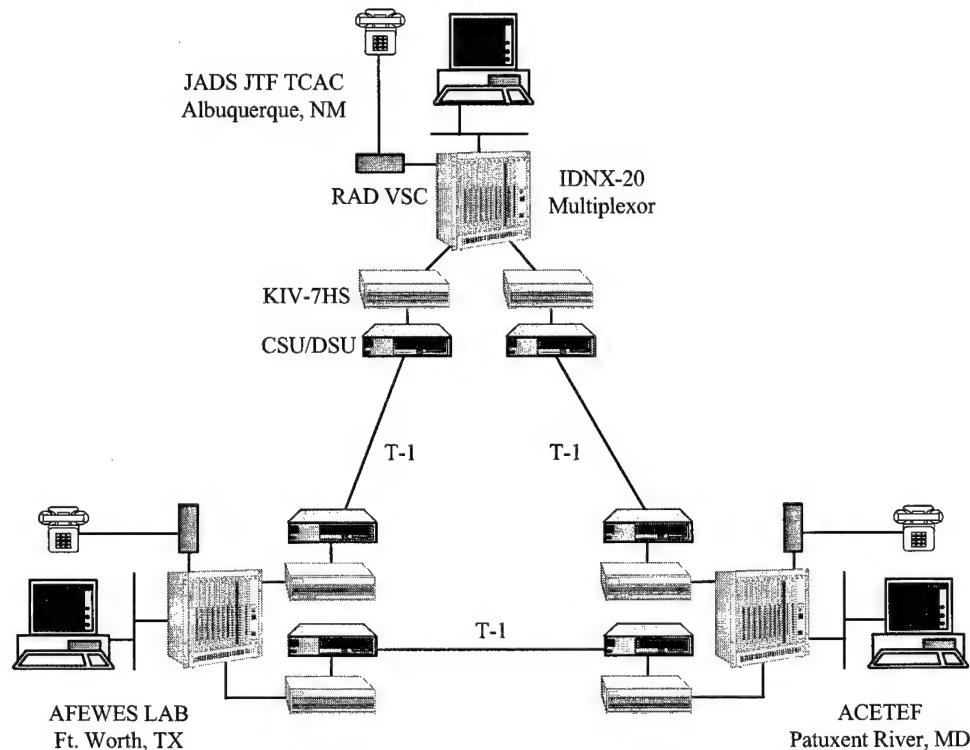
The ROE restricted the use of site operator enhancements such as electronic counter-countermeasures (ECCM), optics, and moving target indicator (MTI) modes. AFEWES operators used the same systems capabilities and operator techniques following the Phase 1 ROE. JADS EW Test team members closely monitored operator implementation of the ROE to maximize collection of useable data and limit variability induced by human error. The aircraft flight path was generated from actual TSPI recorded in Phase 1. This flight path mitigated the background effects of clutter, glint, and multipath on the OAR.

### **2.5.4 Test Configuration**

The JADS Network and Engineering (N&E) team and the EW Test team cooperatively developed the wide area network (WAN) and local area network (LAN) within the TCAC based on cost, resident knowledge, and software requirements. The LAN was composed of the federates residing in the JADS TCAC: platform, TCF, TTH, RFENV, and analysis. The federate software was developed and designed to function in the HLA communicating via the DMSO RTI. The EW Test team used JADS-purchased hardware and software to link the respective locations.

### **2.5.4.1 Wide Area Network Components**

This section describes the equipment installed by JADS at each location (JADS, ACETEF, and AFEWES) comprising the WAN. Although it was not a requirement, it was desirable to utilize commercial-off-the-shelf (COTS) equipment that was easily obtainable and reasonably affordable. JADS procured all the WAN equipment from existing governmental contracts with significant cost savings compared to the vendors' list prices.



CSU = channel service unit

DSU = data service unit

IDNX™ = Integrated Digital Network Exchange

KIV = AlliedSignal embedded KG-84 communications security module

RAD = company that manufactures the voice signal converter

T-1 = digital carrier used to transmit a formatted digital signal at 1.544 megabits per second

VSC = voice signal converter

**Figure 3. Wide Area Network Components**

#### **2.5.4.1.1 Hubs**

Originally, simple, unmanaged (dumb) Ethernet hubs were used to interconnect the various computer workstations and the router at a site into a single Ethernet segment. JADS utilized a variety of unintelligent 10 megabits per second (Mbps) half-duplex Ethernet hubs available from

multiple vendors. For reasons discussed below, JADS had to replace the dumb hubs with Ethernet switched hubs.

#### 2.5.4.1.2 Ethernet Switched Hubs

The Ethernet switched hubs were used in the same manner as the dumb hubs, to interconnect the computer workstations and the router at a site to form a single Ethernet segment. The main difference between a switched hub and a dumb hub is that a switch will selectively route packets between ports, whereas a dumb hub will retransmit all incoming packets to all ports. On a switched hub, only broadcast packets are retransmitted to all ports. Also, switched hubs operate at full duplex while dumb hubs operate at half duplex. Dumb hubs can have many Ethernet packet collisions while switched hubs show few, if any, collisions. The EW Test used SGI workstations running the IRIX operating system. The IRIX transmission control protocol (TCP)/Internet protocol (IP) stack had difficulty dealing with collisions. Collisions on the network would cause large latencies in reliable message traffic. This was avoided by implementing 10/100 Base-T auto-sensing Ethernet switches that were available from multiple vendors. The auto-sensing feature allowed JADS to connect 10 megabit (Mb) (e.g., the routers) and 100 Mb systems to the same Ethernet device.

#### 2.5.4.1.3 Channel Service Unit/Data Service Unit

The channel service unit (CSU)/data service unit (DSU) interfaces the KIV-7HS encryption device or the Integrated Digital Network Exchange (IDNX™) trunk module to the T-1 communications line by converting the nonreturn to zero (NRZ) output of the KIV-7HS to a bipolar alternate mark inversion (AMI) signal for transmission over the telecommunications carrier facilities. In addition, the CSU/DSU supports binary eighth zero substitution (B8ZS) encoding and inserts framing bits in the extended super frame (ESF) format. Also, the model (VERILINK AS2000) of CSU/DSU used by the test networks was capable of remote configuration management and monitoring.

#### 2.5.4.1.4 KIV-7HS Encryption Device

The KIV-7HS is a National Security Agency (NSA)-certified link encryption device that was used to protect the data being transferred between sites. The KIV-7HS protects classified and sensitive digital data transmissions (Type I) at data rates up to 1.544 Mbps. Its performance characteristics are similar to the KG series of cryptographic equipment. The KIV-7HS supports the T-1 data rate with one-way end-to-end latency through a pair of KIV-7HS encryption devices of 4.5 microseconds. The primary reason for using the KIV-7HS was the significant cost savings over the KG series of encryption devices. The cost of installing a pair of KIV-7HS encryption devices on a communications circuit was \$7,969 versus \$20,800 to install a pair of KG-194 encryption devices.

#### 2.5.4.1.5 Integrated Digital Network Exchange

The IDNX is a communications resource manager (CRM) (multiplexer) that supports and integrates a broad range of voice, data, and internetworking services. The entire network was monitored, managed and controlled from any IDNX node in the network. JADS chose the IDNX-20 series of CRM because of these features. The ability to configure and manage the systems from a single location allowed JADS to quickly troubleshoot problems and reconfigure the network equipment. The following subsections describe the feature modules utilized by the EW Test.

##### 2.5.4.1.5.1 I422 Trunk Card

The I422 trunk card provided an RS-449/422 compatible interface for the IDNX to interface with the KIV-7HS or the CSU/DSU (nonsecure applications). The module also contained a crypto-synchronization relay that allowed it to support automatic external resynchronization of encryption equipment. The I422 trunk module did real-time multiplexing, synchronization, internodal signaling and contained the logic to control allocation of trunk channels. It allocated 16 kilobits per second (Kbps) of the T-1 bandwidth to an internodal communications channel that was the sole means by which nodes communicate. The channel carried data that allow the network manager to configure, query, and monitor all nodes from anywhere in the network. The internodal channel provided

- Call processing, configuration, network events, and status information to all nodes in the network
- Code loading when the desired code was not present in the node
- Database information, events, alarms, and circuit management messages to the network manager
- A continuous bit error rate test (BERT) in 30-minute intervals on the communications circuit

##### 2.5.4.1.5.2 Packet Exchange (PX)-3 and Access PX Router Modules

The packet exchange (PX) platform is a general purpose router/bridge module integrated into the IDNX CRM. The PX platform provided packet-switched services between LANs over a WAN through the IDNX CRM. The module connected the TCAC LAN to the WAN via an Ethernet. The PX platform featured an onboard processor and up to eight high-speed serial ports. PX platform serial ports can be connected to remote PX modules or to local or remote data cards with external serial ports. The PX-3 module was implemented for the EW Test because it supports IP multicasting. The PX-3 module utilized Cisco release 11.1 for its operating system. In addition, the PX-3 module supported IP multicasting and was year 2000 (Y2K) compliant.

#### **2.5.4.1.5.3 Quad-Analog Voice Processor Module**

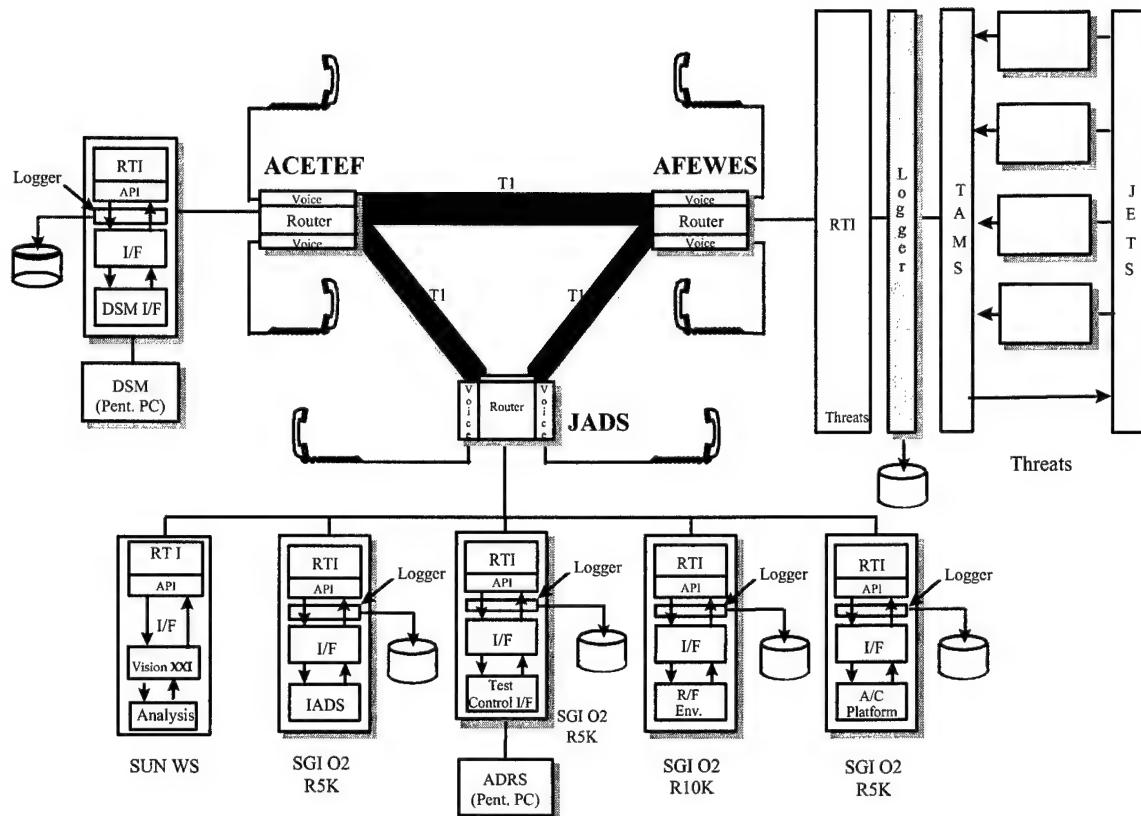
The quad-analog voice processor (QAVP) module provided and managed voice calls coming into and leaving the WAN. It served as the interface between external voice communications equipment and the rest of the network. The QAVP module supported four full-duplex channels, which connected to industry standard 4-wire E&M analog communications equipment. The module converted 3 kilohertz (kHz) bandwidth analog signals to 64 Kbps digital pulse code modulation (PCM) and vice versa. It featured echo cancellation, which eliminated echo caused by hybrid transformers that connected two-wire circuits with analog four-wire circuits.

#### **2.5.4.1.5.4 RAD Voice Signal Converter**

The RAD voice signal converter (VSC) interfaced between an ordinary 2-wire telephone set and the 4-wire E&M interface enabling direct connection to the analog interface of a time division multiplexer. The VSC recognized the telephone set pulses for on hook, off hook and dialing, translated the pulses into the proper signaling standard, and sent the resulting signal over the "M" lead. When detecting activity on the "E" lead, the VSC sent the ring signal to the telephone and the ring back tone to the 4-wire E&M interface of the QAVP.

### ***2.5.4.2 Federate Components***

The set of simulations and the model comprising the JADS EW Test Phase 2 interacted via the services of the HLA RTI in accordance with the JADS EW Test FOM and a common HLA rule set. To illustrate where the RTI resides, Figure 4 depicts the flow of information through the RTI within the entire federation as well as the individual federates.



A/C = aircraft

API = application program interface    I/F = interface

T-1 = digital carrier used to transmit a formatted digital signal at 1.544 megabits per second

**Figure 4. JADS EW Test Federation**

#### 2.5.4.2.1 DSM Federate

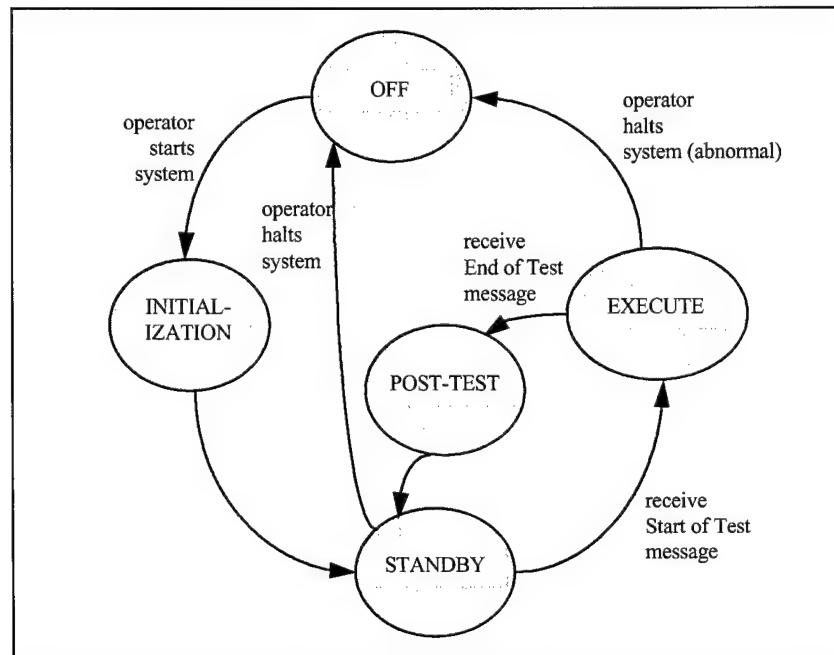
During federation execution, the DSM processed time-referenced environmental inputs of the aircraft location and attitude, threat system mode of operation, and threat antenna pointing (tracking error) data. The DSM received inputs from the JADS RFENV federate (through the JADS federate interface software), recreated the received signal environment for the simulated ALQ-131 R/P, simulated the modified ALQ-131 processing of received threat signals, selected the appropriate jamming response, and transmitted the selected jamming technique to the federation through the JADS federate interface. The DSM operated in four states: initialization, standby, execute, and post-test, as illustrated in Figure 5.

**Initialization** was the initial state of the system after it was started. The initial operation after the creation of the normal federation execution was the entity roll call. As each entity reported, the test director examined the entity's status to determine whether the entity was prepared to proceed with the test. During this time, the DSM provided a status display to the DSM operator and transition into the standby mode.

The **standby** state permitted the test controller to hold selected entities (including the DSM) and signified that the DSM was ready to begin at the chosen start time. When the test controller was satisfied that the federation was ready to proceed with the test, a start command was issued (sent to all federates) and the DSM was placed in the execute state. While in standby state, the DSM sent link health check attributes from the federate and transitioned to the execute mode when the start of test command was received from the test control federate (TCF).

In **execute** state, the DSM processed all inputs from the JADS environment (i.e., other federates), checked for errors, generated outputs to the JADS environment, provided a status display to the DSM operator, and transitioned to the post-test state when the end of test command was received from the TCF.

In **post-test** state, the DSM saved recorded time histories to disk, reinitialized the environment generator (ENVGGEN) and radar processor simulation (RPSIM), and returned to the standby state.

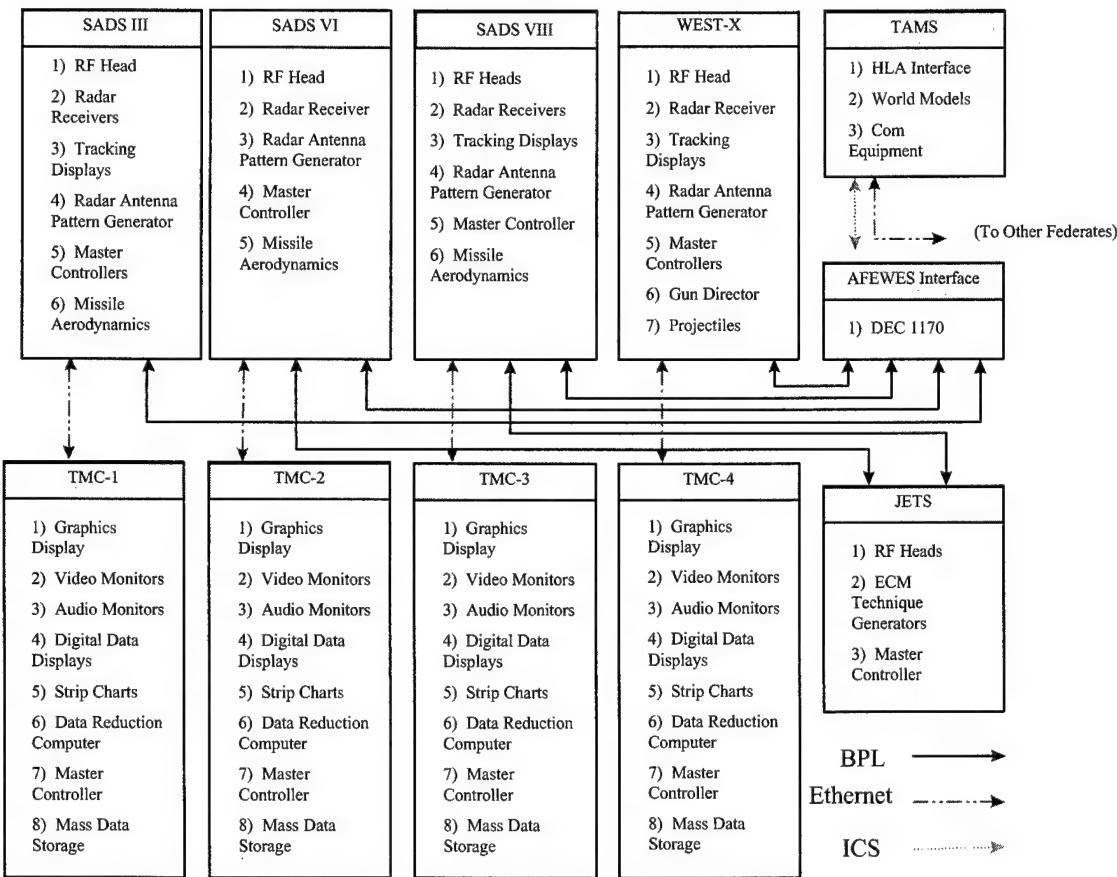


**Figure 5. Jammer/DSM State Transition Diagram**

#### 2.5.4.2.2 AFEWES Threats Federate

The AFEWES facility consisted of multiple man-in-the-loop threat simulations, an internal LAN, sophisticated EW effects generators, diverse computer systems integrating AFEWES capabilities, test management capabilities, and a gateway for linking with external facilities. The external gateway was designed to support the ADS-based testing in Phases 2 and 3. This was the only component of the AFEWES federate that did not exist before the EW Test. All these components

were integrated into a single federate and used to support the test. This federate was responsible for providing the terminal threat simulations. These simulations included human-operated threat simulators designed to track a simulated target. The simulated jammer output from the target aircraft was injected at RF into each threat simulation to provide a realistic EW engagement. The threats were provided a RF simulation of the SPJ technique waveforms using JETS. Figure 6 represents the facility components used during Phase 2. The HLA interface (gateway) within TAMS was the major application residing at AFEWES specifically developed for this test. It permitted communication among AFEWES and the other federates by using ICD-compliant message formats. The logger was another slice of software used. It collected specific federate data processed both in and out of the interface. The AFEWES federate required no other software or hardware additions to facilitate test execution.



**Figure 6. AFEWES Federate Configuration**

#### 2.5.4.2.2.1 Threats

The AFEWES closed-loop, surface-to-air weapon system simulations for this test consisted of an RF head, a tracking console, a software-programmable antenna pattern generator (SPAG)

computer and customized software programs, and a master computer that provided overall real-time control of the target tracking simulation. The RF head simulated real-time echo signals. It also modulated and scaled RF signals to simulate the effects of range, antenna gain patterns, and other factors in the radar range equation. The tracking console contained receivers, target tracking servo-control systems, a system synchronizer, simulation displays, and man-in-the-loop radar operator controls. Antenna patterns were simulated on the SPAG computer through a table look-up process. The simulators used during JADS tests were the SADS III, SADS VI M, SADS VIII, and WEST X. The JADS and ACETEF federates interfaced to the AFEWES simulators through the TAMS computer which hosted the AFEWES threats federate that provided an HLA-compliant interface to AFEWES and each simulator. The TAMS computer is an SGI Challenge multiprocessor system. A Digital Equipment Corporation (DEC) 11/70 was used as an interface buffer among the four simulators and the TAMS. The TAMS data represent the aircraft platform consisting of TSPI, attitude, RCS, and the pod antenna patterns. The AFEWES federate in TAMS took raw simulator output data and formatted them according to the JADS FOM. It also converted input data from other federates into a readable format by the appropriate simulator.

#### 2.5.4.2.2.2 Test Management Centers

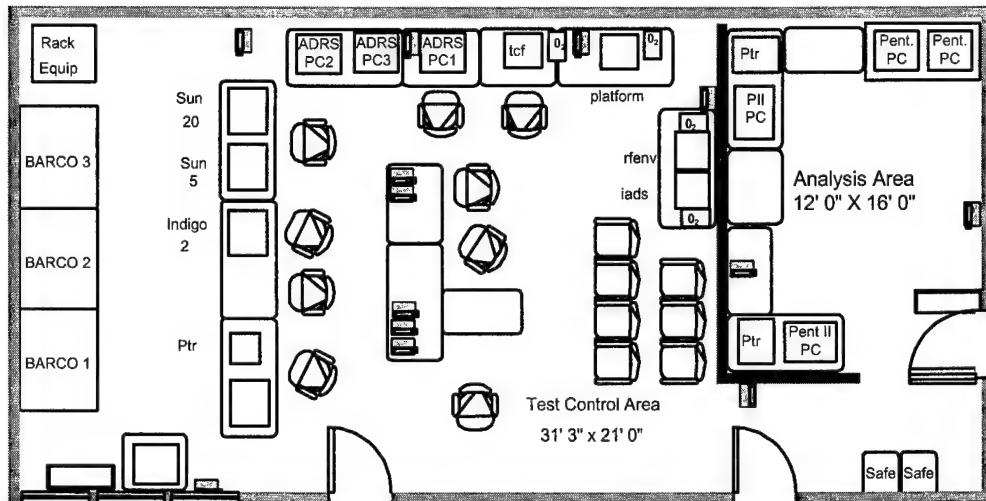
The TMC were used to monitor and collect data such as jamming-to-signal ratio (J/S), radar tracking error, and missile/projectile miss distance from the four closed-loop simulators. There was a separate TMC associated with each simulator. Raw data collected during the test were available for evaluation via digital strip chart printouts, graphics, miss distance calculations, and radar scope videotapes.

#### 2.5.4.2.2.3 JETS

The AFEWES JETS was used to generate SUT ECM techniques to AFEWES simulators for Phase 2. Specifically, during the test runs involving generation of closed-loop engagements with simulations of the SADS VIII and the West X, JETS supplied RF responses representing the SUT emissions. Then during the generation of closed-loop simulations of the SADS III and SADS VI M, JETS supplied representative SUT ECM technique emissions.

#### 2.5.4.2.3 TCAC Test Federates

In addition to the federates located at AFEWES and the DSM located at ACETEF described previously, the five other federates located at JADS in the TCAC (see Figure 7) were the platform, RF environment, terminal threat hand-off, analysis, and test control.



ADRS	Automated Data Reduction System used for test control and post-test data reduction
Barco	large screen display systems used to display various computer screens
Equip Rack	equipment rack with Barco switches, global positioning system (GPS) receiver, tape recorders, phones, etc.
iads	terminal threat hand-off federate
indigo2	SGI Indigo2 used as a network and engineering workstation
O <sub>2</sub>	SGI O2 workstation
Pent PC	Pentium PCs used for post-test analysis
PII PC	Pentium II PCs used for post-test analysis
pltfm	platform federate
rfenv	RF environment platform
tcf	test control federate
sun20	Sun SparcStation20 used to host SPECTRUM®, a network monitoring tool
sun5	Sun SparcStation5 used to host the analysis federate from TRAC Monteray

**Figure 7. EW Test Control and Analysis Center**

#### 2.5.4.2.3.1 Test Control Federate

The TCF managed the test execution and collection of data necessary to evaluate SPJ and ADS performance measures. The TCF interfaced with the ADRS computers serving as a pass-through federate to propagate the setup, start-up, and stop commands as well as performing other test control and display functions. The ADRS setup command initiated loading the corresponding run scripts in the platform, TTH and RFENV playback federates located in the TCAC. TCF transmitted the setup, start-up, and stop commands to all federates (DSM, AFEWES, RFENV, TTH, platform, and analysis) to control the beginning and end of the run. The TCF federate also provided the required HLA interface capability for the PCs hosting the ADRS applications. TCF passed relevant status, performance, and position data for other federates at JADS, AFEWES, and ACETEF to ADRS that provided unique EW Test visualization features of ADRS during each test run. Visualization features included real-time displays of jammer and threat emitter status and a heads-up display (HUD) showing aircraft attitude, altitude, and speed. Specific EW Test performance displays (e.g., J/S) were also provided by the ADRS during each run. Three PCs hosting ADRS were connected to the TCF. Two of the ADRS PCs were used for real-time monitoring by JADS analysts during each run. The third ADRS PC provided a "hot spare" for the analysts in case the ADRS software crashed during a run.

#### 2.5.4.2.3.2 Platform Federate

The F-16 aircraft modeled in Phase 2 (called the platform federate) provided a composite of position and attitude. No systems or characteristics of the F-16 were represented in the model. Aircraft position and attitude for each test run recorded from the OAR TSPI and inertial navigation system (INS) were played back in the form of a data script by the platform federate. For selected threats, the platform federate script also played back threat-centric tracking error measured in the OAR. Since AFEWES operates a SADS VI M and the WTR has a SADS VI M, the data recorded on the range were used to set up the engagement at AFEWES. In order to ensure proper sequencing of SADS VI M tracking data with aircraft position, threat performance data for the SADS VI M target tracking radar (TTR) were also included in this script. The TSPI position and attitude as a function of time had to correspond to a real pass made in the OAR to have maximum value for correlation. Each threat simulator at AFEWES responded to an RF signal that represented a theoretical reflection of the aircraft. The signal was built at AFEWES using the aircraft position and attitude in relation to the threat and “looking up” an RCS value from a table. A four-way power interpolation was performed based on these table values to obtain the correct RCS values for the exact aircraft position relative to the site. The signal was then modified to account for its distance and motion relative to the threat as well as the relationship between the threat antenna boresight and the aircraft position. This is also the way the F-16 RCS was presented during the HITL test.

#### 2.5.4.2.3.3 RF Environment (RFENV) Federate

This playback federate was responsible for reporting emissions of the two unmanned simulators to the DSM/jammer federate as collected on the range. The data were time ordered (as they occurred on the range) and replayed based on an elapsed time from start sent by ADRS through the TCF. In order to replicate the test environment from the OAR test phase during the ADS phases, these background threat emissions were simulated in the ADS test environment as modes for injection into the DSM/jammer federate. RFENV was not programmed to transmit data when AFEWES was operating all four threats. Originally, the RFENV federate concept was not designed for this purpose. It was designed to emulate the extraneous RF emissions and signal distortions recorded during the OAR test. These signals were to be applied to the ADS environment through this federate. Unfortunately, the instrumentation used to record and subsequently recreate the conditions of the OAR test was inadequate for this requirement. Since the data were not available for replication, RFENV evolved into a federate populating the environment—published data for a simulated threat pair—to maintain the appropriate threat density for the DSM.

#### 2.5.4.2.3.4 Terminal Threat Hand-Off (TTH) Federate

This federate was responsible for assigning terminal threats located at AFEWES to the target simulated by the platform federate during an engagement. The test controller at AFEWES received a visual cue on the TAMS to turn on/off the appropriate threats. The test controller transmitted a voice command relaying the on/off cues to the threats sent from the TTH federate.

The time for each cue was taken from the OAR and HITL tests site controller's matrix. This information was scripted and played back sequentially based on elapsed time from the start command. The initial intent of this federate was to act as the command and control element of the threats by transmitting digital commands directly to the simulators; however, AFEWES was unable to support this technical design without modifications to some simulations. The alternative implementation used digital commands sent from the TCAC to the AFEWES gateway via a TTH federate, and then the test controller read the commands off the display to the threat operators. This approach more closely duplicated the voice command structure used during the OAR phase.

#### 2.5.4.2.3.5 Analysis Federate

The analysis federate had many of the same functions of ADRS; however, it took a different approach to data collection and scenario visualization. It was another useful visualization tool for the test controller. It produced a top down view of the scenario similar to ADRS, but also shows threat site modes, and missile flyouts. Other displays on the perimeter showed the EW Test MOPs and measures of effectiveness (MOEs) being calculated as the scenario evolved. It added the benefit of EW Test MOP analysis in real time, and also aided in quality assurance of the data and troubleshooting of anomalies.

### 2.5.5 Instrumentation

#### 2.5.5.1 *TrueTime Global Positioning System (GPS) Receiver*

The GPS receiver is a time source provided by the GPS satellite constellation. It had Inter-Range Instrumentation Group (IRIG)-B, 1 megahertz (MHz), 5 MHz, and 10 MHz signal outputs for use by timing distribution systems.

#### 2.5.5.2 *BanComm Timing Cards*

The computers in the TCAC and at ACETEF all contained BanComm cards that were connected to the IRIG-B time code signal from a GPS receiver. All the federate software and the DSM software executing on the DSM PC obtained time directly from the BanComm cards. The PCs running ADRS also contained BanComm cards. However, because the ADRS software was a Windows 16-bit application and BanComm only had 32-bit drivers, those PCs obtained their time from the PC system time that was periodically set to GPS time by a BanComm utility program.

#### 2.5.5.3 *JADS RTI Interface Logger*

The logger resided in the software interface between the federate and the RTI. It recorded all function calls to and from the RTI along with all the function data parameters. For example, when the federate wanted to publish data, it called the RTI updateAttributeValues function. When the logger was linked with the federate, the federate called the logger updateAttributeValues function. The logger stored the function identification and parameter data

in the log file buffer and then called the RTI updateAttributeValues function. When a log file buffer became full, it was written asynchronously to the log file and a new buffer was created.

The logger was designed to minimize impact on the federate it was linked with. To accomplish this, the logger design included the following features: asynchronous direct input/output (I/O), nondegrading process priority, and binary file format.

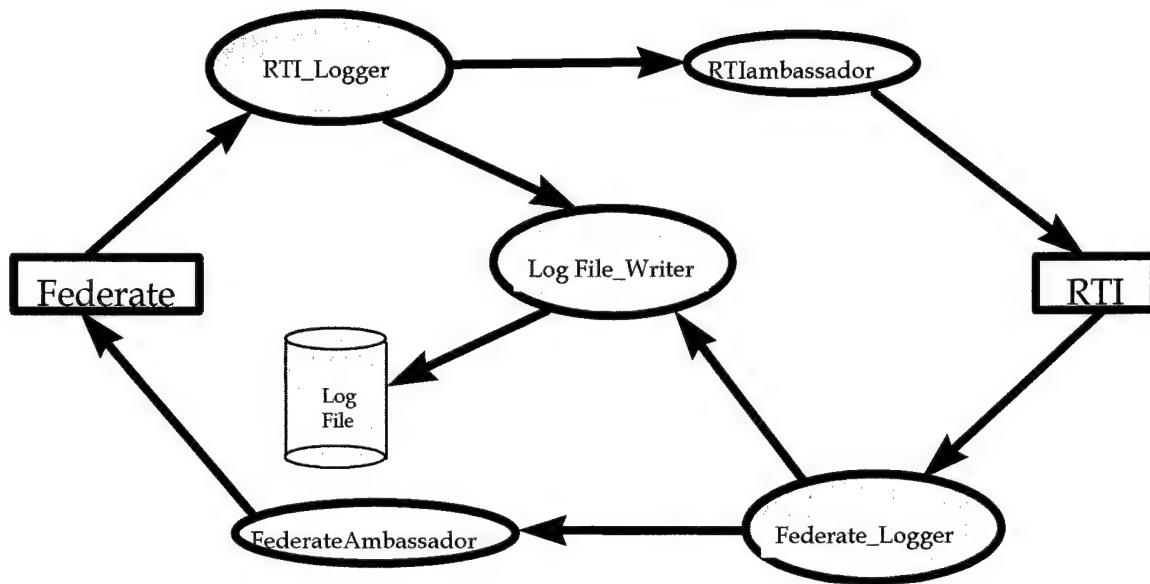
Asynchronous I/O was used so that the federate software did not wait while the data were written to the log file. When a buffer became full an I/O request was queued to the operating system and control was immediately returned to the federate. A separate process accomplished the actual writing of the data.

Direct I/O allowed the operating system to use the data buffer created by the logger software to write the data to the disk. Normally, the operating system copied the data from the user's buffer to a system buffer. However, use of direct I/O eliminated this copy operation.

If a user with super-user privileges executed the logger within the federate, the logger would take advantage of system's nondegrading priorities to further minimize the impact of the logger I/O on the federate. When asynchronous I/O was initialized, a set of processes was created to write the data to disk. When these processes were created, they inherited the priority of the process that created them. The logger software lowered the priority of the process before the asynchronous I/O was initialized. After the I/O processes were created, the logger software set the federate process priority to a real-time priority. Since the I/O processes executed a lower priority than the federate process, the I/O processes never interfered with the federate process.

The log file created by this software was a binary file. Attribute and interaction data were received by the logger (or by the federate) in a binary format. In the interest of minimizing the processing time used by the logger, the binary data received from or sent to the RTI were written directly to the log file without any conversion.

Since the logger software writes all of the binary data sent to or received from the RTI without attempting to translate or convert them, the logger can be linked with any federation without modifications to the logger software. Also, since the logger classes are derived from the RTI base classes, very few lines of code must be changed to incorporate the logger into an existing federate. Less than twenty lines of code were modified or added to link the logger with the helloWorld demo program provided with the RTI.



**Figure 8. HLA Logger Implementation Diagram**

#### **2.5.5.4 Network Monitoring**

A combination of in-house tools and commercially developed software products provided JADS with a real-time, or near real-time, limited capability to assess network performance and evaluate the integrity of data as they were being collected during Phase 2. The various tools were used to help provide a clear picture of the network and to speed diagnostic and maintenance efforts. Data on the network were not collected for post-test analysis during Phase 2. This will be changed for Phase 3 based upon observations made during Phase 2 post-test analysis.

Simple Network Management Protocol (SNMP) tools were used to monitor communications and network hardware. This allowed JADS personnel to see, in near real-time, the status of the long-haul links as well as the routers connecting remote sites. The SNMP tools also allowed JADS to monitor and record the bandwidth used on the T-1 links.

Each federate sent link health check updates and displayed the information received from other federates. This was used during Phase 2 testing not only to determine the health of each federate but as another view of the overall health of the network.

In addition, a simple utility used standard pings to display the status of various federation computers. This tool often presented the first indication that a problem existed with the network and/or the computers at each site.

For some runs, data dropouts and high latency spikes were noticed in post-test analysis. Router statistics were examined in real time (very intrusive) for a few runs to determine if the routers were the problem. There was no indication that the routers were dropping packets and latency could not be examined with the available data. It was determined that protocol analyzers

(sniffers) located on each LAN segment at each node would have allowed JADS personnel to determine the causes of some of the data dropouts and high latencies. Sniffers were not used during Phase 2 because changes to the Ethernet architecture would have been necessary at each site and the required numbers of sniffers were not available. Sniffers will be incorporated into the Phase 3 architecture.

#### 2.5.5.4.1 Network Traffic Analysis

SPECTRUM®, a network analysis package developed by Cabletron Systems, provided a near real-time capability for network traffic monitoring, presented current packet rate and load information, as well as packet error and discard rate information, for network equipment. In addition, the SPECTRUM alarm manager, with simple diagnostic capability, was valuable in the detection and troubleshooting of network outages. SPECTRUM utilized SNMP to periodically query network devices and display requested information on screen in table and graph format. The SPECTRUM operator tailored the destination, frequency, and content of the queries to provide the desired level of insight into a particular network portion or piece of equipment. Typically, a thirty-second polling frequency was used to monitor the equipment. SPECTRUM's event log and query results were also stored to database for post-test analysis.

#### 2.5.5.4.2 Link Availability Monitor

There were numerous ways JADS personnel gained insight into the availability of a particular EW Test network link. For instance, a sudden drop in packet rate picked up by SPECTRUM usually indicated a network link problem. Another solution made use of the self-diagnostic capabilities of the network equipment. A line printer in the TCAC was set up to print diagnostic messages directly from the IDNX multiplexer. The sound of the printer drew immediate attention to a potential equipment outage. JADS programmers coded another simple tool, based on the UNIX ping utility that allowed test controllers to quickly verify link availability with a glance at one screen. Called the "stop-light" tool, it presented a small green, yellow, or red on-screen graphic for each monitored link based on the link's current status. If pings were delayed or dropped status changed. Ping data were not stored for future analysis

#### 2.5.5.5 Network Health Check

For Phase 2, there were two types of periodic federate health checks. The test showed that neither type provided completely satisfactory instrumentation for that purpose.

##### 2.5.5.5.1 RTI Heartbeat

The first health check was the internal RTI "heartbeat" message sent every six seconds via TCP/IP from each federate to the federation executive (FEDEX). If the FEDEX failed to detect three successive heartbeat messages from a federate, it would display a warning message in the FEDEX window on the SGI O2 host RFENV federate.

#### **2.5.5.5.2 Federate Link Health Check**

The federate link health check (LHC) system, as documented in the Phase 2 ICD, was the second health check. This system employed 1 hertz LHC messages sent best effort from every federate to every other federate. It proved to be much more useful, both in real time during the test runs and later during the post-test analysis, because of its higher frequency and the fact that the LHC messages were captured by the JADS RTI logger.

### **2.5.6 Test Control and Monitoring**

#### ***2.5.6.1 Test Control and Analysis Center***

The TCAC located at JADS served as the hub for test control and data collection for Phase 2. In addition to the five federates (platform, TCF, TTH, analysis, and RFENV) residing in the TCAC, the network monitoring, data collection and storage, test visualization and analysis were all performed within the TCAC. The TCAC test controller and operators had voice communications to the two sites, AFEWES and ACETEF, and were able to relay federation commands and speak directly with the site observers over a conference phone system. The TCAC systems provided the test manager with the capability to monitor and control the test execution with the assistance of the site observers and the other federate operators.

#### ***2.5.6.2 Site Observers***

JADS representatives were positioned at AFEWES and ACETEF to observe critical test elements or events. These observers used on-site visualization tools as well as direct observation of operator actions to provide additional insight during test execution and post-test analysis. Observers at AFEWES provided detailed notes of simulated threat actions, JETS operations and the AFEWES threats federate. For each run, the observers noted whether the run was considered usable or unusable for analysis based on the appropriate responses from a particular system. All visible anomalies were noted. This information was helpful and necessary in discerning the quality of usable data during the analysis process. Although detailed information regarding the engagement was readily available at AFEWES in the form of strip charts, the handwritten notes augmented these digital printouts.

### **2.5.7 Runtime Infrastructure Software**

The JADS federation implemented the RTI Version 1.3R4 for SGI O2 computers using the IRIX 6.3 operating system. The federates conformed to Version 1.3 of the HLA interface specification.

### **2.5.8 JADS EW Test Federation Object Model**

In simple terms, a FOM is the identification of the objects, their attributes and interactions. The FOM used by the JADS EW Test was prepared in accordance with the HLA Object Model Template, Version 1.3. The JADS federation implemented for Phase 2 was the JADS EW Test FOM Version 1.0.

### **2.5.9 JADS EW Test Interface Control Document**

The JADS EW Test ICD specified the HLA interface requirements between JADS EW Test federation members at a level sufficient to implement all requisite RTI service calls. The ICD augmented the FOM with information required to develop and implement the JADS EW Test federation. The ICD version used for the JADS EW Test Phase 2 was Version 1.3, dated November 23, 1998. The ICD and FOM were critical to implementing the HLA and executing Phase 2.

## **2.6 Schedule**

This schedule outlined the major tasks and associated execution time windows. The matrix below includes the preliminary actions required prior to Phase 2.

<b>Event Date</b>	<b>Completion Date</b>	<b>Event</b>
2 - 5 Nov 98	1 Dec 98	DSM Acceptance Test
10 - 12 Nov 98	19 Nov 98	Federate Acceptance Test
17 -19 Nov 98	19 Nov 98	Federate Integration Test
24 Nov 98	24 Nov 98	Phase 2 Test Readiness Review (TRR)
1 - 4 Dec 98	4 Dec 98	Execute Phase 2 (SADS VIII/WEST X)
7 Dec 98	7 Dec 98	Execute Phase 2 ADS Excursion (SADS VIII, WEST X, SADS III and SADS VI)
7 -11 Dec 98	11 Dec 98	Execute Phase 2 (SADS III/SADS VI)
1 - 11 Dec 98	14 Dec 98	Quick-Look Report (daily)
14 Dec 98	18 Dec 98	Quick-Look Report (summary)

## **2.7 Security**

The highest classification level of data processed by the EW Test was secret/US only. The highest level of data reported was secret. To ensure the proper classification of data collected and presented, the team incorporated classification standard operating procedures and information

security policy from elements directly related to the test. Additionally, the SPJ security guide was used.

### **2.7.1 Network Security**

Phase 2 established both secure voice and secure digital by using KIV-7 encryption devices throughout the WAN. These devices permitted point-to-point transfer and verbal transmissions of classified information. Once JADS and AFEWES and JADS and ACETEF reached their respective security agreements, the WAN segments were operationally ready to handle classified data.

### **2.7.2 Data Security**

JADS signed formal security agreements to pass information up to the security classification of secret across the network. However, most of the data transmitted from node to node were unclassified. Data exchange, safeguarding and labeling were commensurate with security classification guides and policies established by the governing agencies.

## **3.0 Preliminary Testing Events**

Developing the components for the EW Test, integrating simulators, software and test facilities, and implementing the network architecture were an incremental process. Preparations began when the JADS EW Test was chartered in August 1996. The following April the test approach was baselined to include the use of the emerging HLA rather than the established distributed interactive simulation (DIS) protocols. In December 1997, JADS built a network test bed and began the first of many test activities focused on the computer, communications, and HLA RTI software supporting the ADS-based test phases. JADS worked closely with DMSO during the development of software (e.g., RTI versions) and tools for federation documentation (e.g., federation execution planners workbook, object model developers tool kit). JADS took initial delivery of key federate software components from GTRI in August 1998 and began stand-alone testing and integration of the federate software. The final software components were delivered in November 1998, and acceptance testing was completed on all the federates comprising the Phase 2 test. This section describes the preliminary events supporting Phase 2 development.

### **3.1 Phase 2 Development Tasks**

To ensure JADS was fully prepared and ready to accomplish Phase 2, a set of tasks with specific completion requirements and schedules was identified as risk reduction steps which preceded the Phase 2 test readiness review milestone. These tasks were

- Network testing
- Test bed development
- RTI performance assessment
- Phase 2 integration
- DSM acceptance test
- Federate acceptance test (FAT)
- Federation integration test (FIT)

### **3.2 Network Testing**

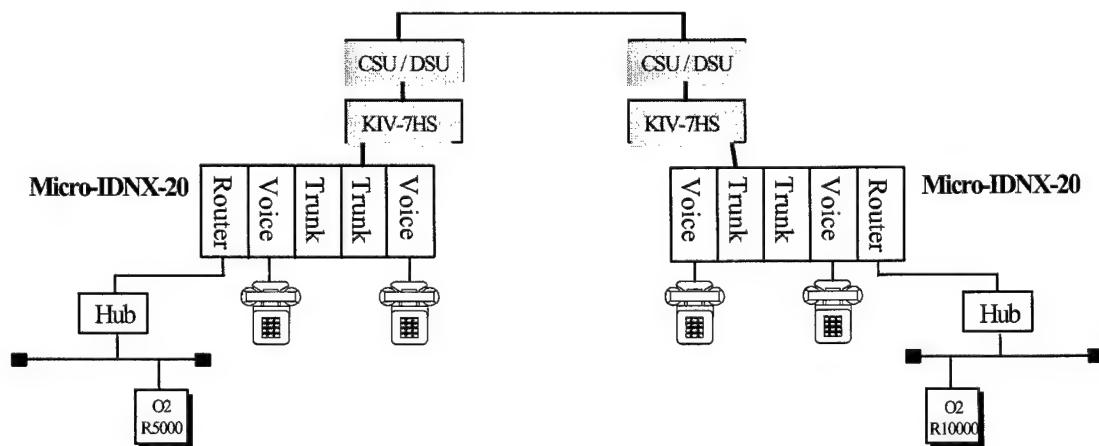
Before testing the RTI software, the network needed to be characterized in the simplest form. To determine the raw network throughput performance, software was developed to send data one-way from one computer to another. Versions of this software perform tests using two data transport modes: TCP and IP with multicast. The one-way software was designed to exercise the network with different data packet sizes and transmission rates. A complete matrix of rate and size combinations was tested. Each test case, defined by a specific rate and size pair, was conducted for thirty seconds. The one-way raw network test consisted of two programs – a sender and a receiver. At the start of each test case, the sender transmitted a start message to the receiver indicating the size, rate and total count of messages to be sent. This information was used by the receiver to name the output file and to determine if any messages were lost. After sending the control message, the sender transmitted the data. The data packet contained a

sequential serial number and the time the message was sent (i.e., when it was time tagged in the sending code). When a message arrived at the receiver, the system time was obtained. To eliminate its effect on the latency calculation, no I/O occurred while the data were transmitted. The data file contained a record for each message that should have been received. If the message was received, the serial number, sent time, received time, and latency were written to the file. This sequence was repeated for every combination of size and rate.

### 3.2.1 Test Bed Development

The RTI test hardware configurations progressively increased in complexity until the entire federation and network architecture (except for the T-1 lines) were in place in the JADS test bed. Starting with a two computer point-to-point configuration shown in Figure 9, JADS Network and Engineering gathered basic performance data for network IP multicast data and network TCP data. Software testing was performed on the following RTI software and data types: RTI 1.0-2 best effort data, RTI 1.0-2 reliable data, RTI 1.3 beta (1.3b) best effort data, RTI 1.3b reliable data, RTI 1.3-2 early access version (RTI 1.3-2EAV) reliable data, and RTI 1.3-2 (early official release) reliable data.

The test configuration included all network components using a two-node network for the same series of tests. The associated communications link and hardware/software configuration were also tested. All sources of possible latency were computed through a disciplined process of adjusting one variable at a time and collecting recorded time data for the same message type transaction in differing reference test conditions. The two-node network test used an SGI O-2 5000 and an SGI O-2 10000 running IRIX 6.3. The test software and RTI were hosted on each computer for all tests using this configuration.



**Figure 9. JADS 2-Node Test Bed Configuration with Communications Devices**

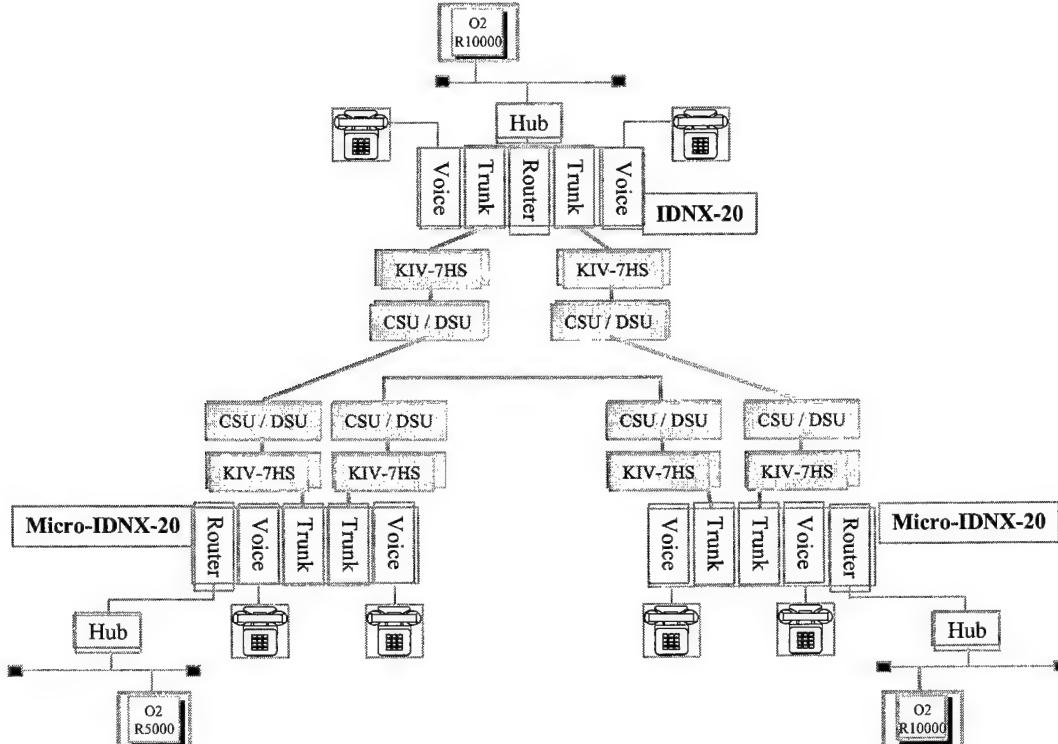
### 3.3 RTI Performance Assessment

After characterizing the network in the simple one-way tests, JADS Network and Engineering needed to determine if the RTI would support the anticipated loads placed on it by the JADS federation. The *testfed* federate was developed to simulate these kinds of loads. The *testfed* federate accepted command line arguments that specified the characteristics of an instance of the federate. The user specified the federate identification (ID) number (-f), the duration of the test (-d), the size of the attributes and interactions (-s), the rate that attributes were published (-r), the number of updates at the specified rate (-n), the amount of time the federate should wait before starting to publish at its specified rate (-w), and whether interactions should be published (-i). There was an additional argument (-c) that indicated which federate was the controller (there could be only one controller federate in the *testfed* federation). There was only one attribute and one interaction which all federates subscribed to. The test was conducted by running with three federates residing on separate computers. For the three-federate test, the *testfed* was configured on one computer to publish 11 attribute updates at 20 hertz (Hz) (simulating the AFEWES federate). Another instance of *testfed* was configured to publish two attribute updates at 20 Hz (simulating the federates at the JADS Albuquerque node). The third instance of *testfed* was configured to publish one attribute update at 20 Hz (simulating the ACETEF node). All three federates published interactions at approximately 1 Hz. The size of attributes and interactions was 121 bytes. Attributes were published best effort and interactions were published reliable. The test team executed multiple tests of between two and five minutes. After the three-federate tests, six-federate tests were run using six computers that more closely resembled the Phase 2 configuration. The tests identified some problems. These problems were fed back to the DMSO technical support for analysis. At the same time, JADS network and analysis personnel analyzed the problems. In some cases the problems were in the network and/or federation configuration. In these cases, DMSO provided recommendations to correct the problem. In the cases where problems were in the RTI, fixes were implemented in subsequent RTI releases. As a new version of the RTI was released, JADS personnel would exercise the RTI with the *testfed* software. Through this process, JADS learned invaluable information about using the RTI, provided feedback on problems and improvements to the developers, and ultimately gained confidence that the RTI would support the Phase 2 performance requirements. Table 4 lists the versions of the RTI that JADS tested prior to Phase 2 execution. RTI version 1.3-4 was the RTI version used in Phase 2 test execution.

**Table 4. RTI Versions Tested by JADS**

<b>RTI Version</b>	<b>Date Released</b>
1.0-2	February 1998
1.3b	3 April 1998
1.3-2 Early Access Version (EAV)	15 May 1998
1.3-2	15 June 1998
1.3-4	October 1998

The test environment expanded from the two-node configuration and used at least three and as many as six SGI O-2 workstations (either R5000 or R10000 models) running IRIX 6.3. The three-node test configuration in the test bed with three SGI computers is shown in Figure 10. Once the RTI performance baseline was determined, further testing, integration, and tuning of all federation components was performed.



**Figure 10. JADS 3-Node Test Bed Configuration**

### 3.4 Phase 2 Integration

The integration of the hardware, software, and facility unique components (e.g., AFEWES HITL simulators, gateways) in preparation for testing was divided into several tasks intended to incrementally assemble and test the Phase 2 architecture. Once the initial network architecture was configured and the benchmarks for the network and RTI were completed, JADS began integration, testing, and build up of the federation components. The major tasks representing Phase 2 integration were DSM acceptance testing (DAT), federate acceptance testing (FAT), and federation integration testing (FIT).

### 3.5 DSM Acceptance Test (DAT)

Formal acceptance testing was not originally planned as part of the development process. However, it became obvious that this testing was needed. Acceptance test procedures were developed to demonstrate that the DSM developed for Phase 2 was an adequate representation of the modified ALQ-131 Block II SPJ pod used in Phase 1 testing. The DSM outputs were used to

study how ADS affected the EW Test results. As such, key characteristics of the DSM had to match the real ALQ-131 characteristics to avoid confusing differences between the DSM and the ALQ-131 with ADS-induced differences. Because the DSM was so critical to the EW Test, JADS determined that this formal acceptance test was necessary.

The DAT was planned to show that the DSM subsystems satisfied certain critical design requirements as documented in the GTRI Software Requirements Specification (SRS) and the EW Test ICD and functioned correctly when used during the FAT, the FIT, and Phase 2. The test was conducted by GTRI and witnessed by representatives of JADS. Discrepancies identified by JADS were addressed by GTRI and resolved during the FAT.

Limitations of the test environment precluded full acceptance test execution. Instead, limited stand-alone testing in a simplistic environment was used to gain confidence that the DSM would meet minimal requirements. The DSM initially passed acceptance tests but several problems were noted during integration and prefederate acceptance tests activity. Further problems were identified during test execution and after data analysis. The DSM, as a federate, was not recommended for accreditation or as a JADS legacy product. The behavior of the DSM was sufficiently like the modified ALQ-131, however, that SMEs recommended overall accreditation of the federation for JADS purposes. Specific shortfalls included failure to disengage then re-engage the same threat, limitations on the number of mode changes that the DSM could process on each run, and failure to generate a range of threat ID and response times. These shortfalls were generally avoided through the scenario, did not affect JADS MOPs, or could be addressed through post-test analysis.

### **3.5.1 DAT Results**

The functions and critical capabilities of the DSM used in Phase 2 are described below. Each requirement was addressed by the DAT.

#### ***3.5.1.1. Received Power Calculation Traceability***

An attempt was made to ensure values in the databases used in the received power calculation were correct or, as a minimum, at least reasonable. JADS accepted that GTRI accomplished this procedure for each threat and the Block II SPJ pod by inspection of the original data sources and comparison of data in the databases to data in those sources.

#### ***3.5.1.2 Comparison of DSM Model Results to an Independent Calculation***

Numerical values for the received power the DSM PC subsystem produced were correct and accurate when compared to those produced by an independent calculation. They demonstrated quantitatively that the received power values calculated by the DSM model agreed to within 0.5 decibels (dB) of those produced from the same input data and equations by an independent calculation, and that the model calculation was sufficient to correctly activate the jamming response from the model. GTRI used four, single-point test cases. The latitude, longitude, and

altitude for each were computed from the initial activation points for each threat from the ROE. The received power was computed by an independent method and by the DSM.

### ***3.5.1.3 Timing Calibration and Traceability***

GTRI presented data intended to demonstrate that part of the DSM PC subsystem correctly simulated the timing responses the modified ALQ-131 Block II SPJ pod generated for each threat radar. The GTRI test configuration did not support transmission or receipt of the external messages. GTRI demonstrated that all databases used by the DSM pertinent to the timing calibration were correct. GTRI accomplished this procedure for each threat and the SPJ pod by inspection of the original data sources, the AMES data, and comparison of data in the databases to data in those sources and from the AMES test. Federate integration testing did not test DSM response time. Post-test analysis showed that the DSM failed to generate more than a few values for timing response for each threat because it was executed differently during Phase 2 than during the federate acceptance test. In particular, the random number generator seed used in the DSM to draw samples from SIL-generated response time distributions was always reset to the same initial value during Phase 2 test execution because of termination of the DSM at the end of each run. During FAT, the DSM was configured to execute multiple runs to demonstrate that it would generate the correct response time distributions. During test execution, it was impossible to run the model continuously because of the limited number of mode changes the DSM could process. This is discussed further in section 4.2.2.1.

### ***3.5.1.4 Real-Time Received Power Calculation Demonstration***

The DSM PC subsystem's received power calculation was correct and reasonably accurate when performed at the full rate required by Phase 2. The simulated jammer activation and the received power calculation results were reasonable for each threat from a real-time demonstration at the required rate compared to the static test case results. Equations that the DSM PC subsystem used were correct and accurately computed the power the simulated SPJ pod would receive from each threat radar in the context of Phase 2. The equations correctly accounted for all factors pertinent to the level of accuracy required. It was also verified by qualitative demonstration that computational procedures and approximations used in the software's implementation of the equations (e.g., discrete approximations for the antenna patterns) did not significantly degrade the accuracy of the received power calculation. The demonstration recreated one slice of one threat antenna pattern by using the tracking error to sweep the threat antenna across the stationary aircraft. Subsequent execution of the DSM PC subsystem uncovered no additional problems with this function.

### ***3.5.1.5 Real-Time Timing Calibration Demonstration***

GTRI demonstrated that the core of the DSM PC subsystem's timing calibration was correct and reasonably accurate. The results for time to correct ID, time to correct response, and age out time (SADS VIII only) from the real-time demonstration with the full rates and required timing were within the population for each threat for five test runs.

### ***3.5.1.6 Correct ID of Threat Demonstration***

GTRI demonstrated that the DSM PC subsystem achieved correct threat ID when used in their limited test environment.

### ***3.5.1.7 Correct Response Mode Demonstration***

GTRI demonstrated that the DSM PC subsystem achieves correct response mode when used in their limited test environment. The DSM PC subsystem selected the correct response mode for all threats and threat modes during the test.

### ***3.5.1.8 Correct Prioritization of Simultaneous Threats***

The DSM PC subsystem selected the correct priorities when used in real time for an engagement similar to one planned for Phase 2. The DSM PC subsystem correctly prioritized responses during the engagement.

## **3.6 Federate Acceptance Test (FAT)**

The FAT demonstrated that the five HLA federates GTRI developed and used for Phase 2 and Phase 3 were adequate to conduct those tests and to compare data from them to the results of Phase 1 tests. The GTRI federates outputs were used to study how ADS affects test results. As such, key characteristics of the federates had to match the characteristics of the systems or environments they simulated to avoid confusing differences between the federates and those systems/environments with ADS-induced differences. Because correct, timely functioning of these federates was critical to the EW Test, JADS determined that a formal acceptance test was necessary.

The FAT served as the V&V test procedure for the GTRI federates and their subsystems. Its scope did not include a sixth federate developed by AFEWES, but the FAT completed some interoperability tests of the GTRI and AFEWES federates.

The FAT consisted of four test phases. The first phase included a series of formal events that verified readiness of all component subsystems for the actual FAT tests. The second phase performed simple tests on each GTRI federate. The third phase tested each federate in a dynamic environment in which the HLA RTI and other messages were exchanged. In the final phase, the testing verified repeatable, robust, and proper functional performance in a dynamic environment in which the GTRI-developed federates exchanged messages, some of which were sensitive to latency. During the last three phases, JADS collected sample data pertinent to various ADS measures.

### **3.6.1 FAT Results**

The FAT showed that the two pass-through federates, through which the non-HLA-based ADRS PC subsystems and the DSM PC subsystem communicated with the EW Test federation, and the three playback federates, which played back time-sequenced data in script files, satisfied specific critical design requirements. In addition, the FAT provided evidence that the five GTRI federates would function correctly when used during the FIT and Phase 2.

## **3.7 Federation Integration Test (FIT)**

The five-phase federation integration test was conducted at JADS, ACETEF, and AFEWES from 17 - 19 November 1998 and provided results required to support the Phase 2 test readiness review and test. The FIT duplicated the Phase 2 test and used two of the four manned simulators at AFEWES (SADS III, SADS VI M, SADS VIII, West X) at a time in addition to federates at JADS and ACETEF. A predetermined number of Phase 2 test runs were conducted daily based on scheduled test events. The FIT was preceded by detailed software acceptance testing (DAT and FAT) addressing the software functionality and adequacy. The FIT focused primarily on the functionality or adequacy of the integrated software capability. The FIT verified the integration and performance of the complete Phase 2 test architecture. The procedures for the FIT included static and dynamic tests organized under five test phases.

- Phase 1 - Network components and time synchronization verification
- Phase 2 - Federation components and functionality
- Phase 3 - Test control and monitoring capabilities
- Phase 4 - Federation execution with manned threat pairs for northbound and southbound runs (wet and dry)
- Phase 5 - Data collection, retrieval, and analysis capabilities

### **3.7.1 FIT Results**

During the FIT, all test start-up, execution, and stop procedures were verified. Site coordination and status control were exercised. Federate and federation execution were fully demonstrated and verified. AFEWES federate software was fully exercised, and integration with GTRI federate software was proven. The 84-run goal for the FIT was not achieved because of problems within AFEWES, but all threats were demonstrated in pairs (SADS III, SADS VI M and SADS VIII, WEST X) and simultaneously. AFEWES internal quality control data were compared to the Phase 1 HITL test run data to verify repeatability. Post-test data processing and analysis capability were performed and verified. Full analysis was not possible because the ADRS software was not calculating all MOPs, but this did not impact test execution readiness.

### **3.8 Verification and Validation**

The EW Test team had two opportunities to accredit the Phase 2 environment. The first opportunity occurred pretest and focused first on the federates and then on the federation as a whole. The second opportunity used the correlation of the results from the OAR test and from the HITL test to determine if the results from the federation were valid. The results of the Phase 2 FAT and FIT were presented to the accreditation board prior to the JADS test readiness review. Issues were raised that caused the board to recommend not accrediting the federation. In particular, instances where the aircraft appeared to hover were reported. The cause could not be established, and the board was concerned that other issues might be present that would make the data invalid. The accreditation authority gave the EW Test team a chance to remedy or address the hovering aircraft issue and look for other issues before executing Phase 2 as scheduled. The hovering aircraft was addressed and no other issues found. The federation was accredited. The V&V plan had to be reaccomplished after Phase 2 execution to satisfy the accreditation board.

Accreditation of the AFEWES threats consisted of a documentation search and key personnel interviews to determine limitations that could impact the JADS effort. JADS was able to find only limited accreditation information on the AFEWES threats. Accreditation information was directed at a threat baseline established through the intelligence community. Similarly, the OAR threats were accredited, however, distinct differences in performance between facilities were observed. These differences ranged from variations in the radar tracking loops to results of threat operator actions. These differences coupled with differences in operator performance precluded attaining good statistical correlation between Phase 2 and the OAR.

Two critical criteria were examined during the acceptance test of the DSM. The first criterion was to determine whether the model correctly implemented several databases to calculate received power. A demonstration comparing the model's calculation with a hand calculation for the same scenario was used to satisfy this criterion. The second criterion was to check that the DSM could be calibrated to produce results within an 85 percent confidence interval when compared to the data used for calibration. This was demonstrated by correlating calibration data distributions with the results from a significant number of model runs. A random number generator was used during this process to obtain a realistic range of received power values. It was discovered after the federation was accredited that the random number generator had been incorrectly implemented given the way the model had to be used to overcome the buffer limitation. The impact was that the results consisted of only a single value within the expected range of received power values. This might have been discovered earlier had the DSM been supported as originally planned. The DSM operator was not from GTRI and had not been involved in Phase 1. The operator had been quickly trained by GTRI to run the model as GTRI support was reduced to stay within cost. This change in plan prevented re-examination of the model after it was integrated into the federation.

The nonmodeling and simulation software was not independently accredited but underwent V&V and was accredited as part of the federation. The most critical components of this software were the federate interfaces and scripted simulations. The interface simulations were the HLA-

compliant interface for the DSM and the HLA-compliant interface for the test control software. In each case the EW Test team used an acceptance test to verify that the interface simulation and the scripted simulation produced and consumed the correct data in the correct format at the correct rates, and that each interacted correctly with the other. The script generator was tested to verify that it produced correct scripts from OAR or HITL data. Once JADS determined that the scripts were correctly generated and the correct scripts were uncorrupted by the simulation, the software was accepted.

The results of these V&V activities were reported to an accreditation board comprised of senior personnel on the JADS JTF with a recommendation to accredit the environment. The accreditation board was concerned about the hovering aircraft anomaly noted in three runs during the FIT. This concern was expressed in two ways: (1) the test team did not know the cause for this anomaly, and (2) the test team could not guarantee that it had only occurred in three runs. Therefore, the accreditation board refused to recommend accreditation of the environment until this anomaly could be explained.

During the test readiness review for Phase 2, the decision was made to proceed with the test without accreditation (the test could not be delayed because of the need to start Phase 3 testing immediately after Phase 2). The test team was directed to spend the time between the review and the beginning of the test determining the cause and impact of the hovering aircraft anomaly. This requirement was met prior to execution of the test, and the anomaly was not observed during the actual test. The cause was bursts of data dropouts from the platform federate. AFEWES implemented a simple dead reckoning algorithm to correct the problem. Prior to executing runs for record, the federation was accredited for JADS. The DSM federate was not accredited, however SMEs felt that the overall federation was acceptable and sufficiently like the expected performance that federation accreditation was still warranted.

## **4.0 Test Execution**

Following the successful completion of development, component testing, acceptance, integration, and verification activities, the EW Test team completed a successful test readiness review at JADS. Based upon the work accomplished, results achieved, and lack of schedule alternatives, Phase 2 was approved for execution.

### **4.1 Test Control**

Test control is an important aspect of any formal test environment. To ensure that test conditions and status were monitored and maintained in a distributed, ADS-based test environment, JADS had to carefully define its requirements for test control, develop unique real-time status reporting and display capabilities, and coordinate and rehearse test control procedures both within the TCAC and with AFEWES and ACETEF. Critical components of Phase 2 test control included federation time synchronization, start-up, and status monitoring.

#### **4.1.1 Federation Time Synchronization**

All computers used for Phase 2 had to use a common time source to ensure valid time values were recorded in the logs at all sites during test execution. Prior to the start of daily testing, a time synchronization test was run by the TCAC to verify that all computers were using the correct, accurate time source (IRIG-B) in their operations and not running on local internal system time. All JADS federates except AFEWES and the analysis federate participated in the time synchronization test. The analysis federate only read and recorded data time synchronized data from other federates. Time synchronization was performed by initiating a normal run and after a jamming response was observed in the TCAC the execution was stopped. JADS would then copy the TCF log file to the appropriate log file directory and run the log file summary program. The recorded times for all messages in federate log files should look reasonable (within +/- 20 milliseconds of each other). The times recorded on the link health check messages showed the time on all of the SGI O-2s. Execution control messages (start and stop) showed the time for ADRS 2 and ADRS 3. The time recorded on the *X\_File\_Spec* message showed the time for ADRS1. The time recorded on the SUT messages showed the time for the DSM PC.

The AFEWES computers were also synchronized to an IRIG time source; software synchronized the system time to the IRIG time. The AFEWES federate software used the system time as its time source. To determine if AFEWES was synchronized, JADS would run a raw TCP test program used previously for RTI testing. Upon completion, the receiver printed out the minimum, maximum, and mean latency. JADS verified that the latency was reasonable. A real-time analysis capability was not available to determine if all systems maintained time synchronization during test runs. However, during subsequent data analysis, the quality of time synchronization could be determined.

### **4.1.2 Federation Start-Up**

During the integration tests, it was determined that there were less data dropouts if the platform federate was not the first federate to join the federation. Since five federates in the TCAC were all subscribing to most of the same reliable data, DMSO technical support recommended that JADS execute with a single reliable distributor for the TCAC. Normally, each federate contains a reliable distributor to send and receive reliable data. There is a TCP connection from every reliable distributor to every other one. By configuring the TCAC federates to use one reliable distributor, the number of TCP connections and the subsequent traffic on the WAN were reduced. The RFENV federate did not have a lot of data to process, so it was chosen as the location for the TCAC reliable distributor. The federate containing the reliable distributor must start the federation executive (FEDEX) and join the federation first. It was observed that if the FEDEX was not started on the same computer where the RTI executive (RTIEXEC) was running, it would take a few minutes for the FEDEX to find the RTIEXEC.

After the RFENV federate started and joined the federation, the other federates in the TCAC would join (staggered by a second or two). After the TCAC federates successfully joined the federation, the remote federates (AFEWES and DSM) would join.

### **4.1.3 Federate Status Monitoring**

The link health check display as well as the FEDEX window was monitored during federation execution. If there were problems, generally one of these windows displayed an indication. For example, the link health check status display on the TCF screen indicated federate status as red or green. It provided the capability to monitor specific multicast traffic paths in one direction only between any two federation nodes (e.g., JADS - AFEWES). The FEDEX window monitored the RTI's "heartbeat" messages over the reliable (TCP) data paths. The federate would display red if the federate stopped sending link health messages because of software error or a failure in the link. However, the problem was generally noted by one of the remote sites before the TCAC federates exhibited a corresponding display. This was due to the fact that the TCAC had fewer indications of information outages from other nodes. The link health and FEDEX windows could only detect outages that occurred over an extended period of time (> 3 seconds for link health, 20 seconds for FEDEX). Status monitoring procedures of individual federates and operator procedures are described in further detail.

#### ***4.1.3.1 Digital System Model Operation***

The ACETEF operator or JADS observer at ACETEF performed the DSM operation as the nominal system under test. Cost constraints prevented GTRI from providing on-site support. The work consisted of two activities performed in sequence for each run.

The first was preparing the DSM federate to join the federation (when directed to do so by the JADS test controller). New directories were created on the DSM O2 and DSM PC to store files the run created, and a new DSM operator's log sheet was started. The *dsm.sh* UNIX shell script was executed on the DSM SGI O2, the run number was entered, and the DSM operator then

waited until the DSM federate link health display stabilized. At this point, the DSM PC software could be started without the risk of causing a core dump of the DSM federate. Proper operation of the DSM federate was confirmed by examining the link health display and the time displays. When requested, the DSM operator reported via the voice link the jammer state for some runs (i.e., transmitter on for wet runs or transmitter in standby for dry runs), but normally only described in real-time mode changes to the DSM PC threat display. This display showed the active threats and whether or not and how the DSM was jamming them. Problems with the federate link health (at the DSM node), DSM federate crashes, etc., were also reported by voice link and logged.

The DSM operator also monitored the DSM PC error displays for missing and out-of-sequence live entity state (LES), source mode change, and threat performance messages. Because of a problem with AFEWES numbering the threat source mode change messages, only the LES and threat performance provided meaningful information. A sudden, large jump in the number of missing LES or threat performance messages was generally a good indicator of either a loss of these data to the DSM federate or of a time synchronization problem with an ADRS PC.

The final activity was to shut down the DSM PC software at end of a run, move the files created by the federate software on the DSM PC and DSM O2 to run-specific directories, and finish the DSM operator's log for the run. For each run, the DSM PC produced 28 American Standard Code for Information Interchange (ASCII) files, while the DSM O2 produced three. Typically, these files consumed several megabytes on each system; however, for some runs, the DSM PC files required many tens of megabytes of storage.

#### **4.1.3.2 Test Control Federate (TCF) Operation**

Executing the *tcf.sh* shell script started the TCF located in the TCAC. When the federate started, it prompted the operator for a run number. After a run number was entered, the federate prompted the operator to join the federation. When directed by the JADS test controller, the operator entered a 'y' at the prompt so that the federate joined the federation. When the TCF had completely joined the federation (as indicated by "continue execution" displayed in the federate window), the ADRS software was started on each of the three ADRS PCs. PCs were started a few seconds apart because if they were started simultaneously, the TCF federate would periodically experience a crash. Once the ADRS software was running, the federate would wait for execution control commands from ADRS. When an execution control attribute was received, it was published by the TCF federate for the other federates. This was the method by which the script names and the start and stop messages were sent to all the federates. When the command was an execution control attribute with an execution control word that indicated stop test execution, the TCF federate published the attribute then resigned from the federation.

#### **4.1.3.3 Platform Federate Operation**

The platform federate located in the TCAC was started by executing the *platform.sh* shell script. When the federate was started, it prompted the operator for a run number. After a run number was entered, the federate prompted the operator to join the federation. When directed by the

JADS test controller, the operator entered a ‘y’ at the prompt so that the federate joined the federation. The platform federate waited until it received an *X\_File\_Spec* attribute update that contained the name of the script to be loaded. When the script name was received, the federate would display the name of the script that was being loaded. Upon completion of script loading, the federate displayed “done.” The federate would then wait for an execution control attribute update with an execution control word that indicated start test execution.

During a federation execution, the platform federate executed without operator intervention. The window in which the federate executed was monitored for error messages. The platform federate played its script until an execution control attribute update was received with an execution control word that indicated stop test execution. The federate then resigned from the federation.

#### **4.1.3.4 Radio Frequency Environment (RFENV) Federate Operation**

The RFENV federate located in the TCAC was started first because it created the federation execution (FEDEX) and the reliable distributor (reldistr) used by all federates in the TCAC. Executing the *rfsenv.sh* shell script started the *RFENV* federate. When the federate was started, it prompted the operator for a run number. After a run number was entered, the RFENV federate created the FEDEX. The federate then prompted the operator to join the federation. The operator entered a ‘y’ at the prompt so that the federate joined the federation. The RFENV federate waited until it received an *X\_File\_Spec* attribute update that contained the name of the script to be loaded. When the script name was received, the federate would display the name of the script that was being loaded. Upon completion of script loading, the federate displayed “done.” The federate waited for an execution control attribute update with an execution control word that indicated start test execution.

During a federation execution, the RFENV federate ran without operator intervention. The window in which the federate executed was monitored for error messages. The FEDEX window was monitored for error messages indicating lost contact with other federates. The RFENV played its script until an execution control attribute update was received with an execution control word that indicated stop test execution. The RFENV federate waited for all other federates to resign from the federation. Once all the other federates had resigned, the RFENV federate resigned from and destroyed the federation execution. If there were problems with any federate resigning, the federation was destroyed manually by entering a “kill” command in the FEDEX window.

#### **4.1.3.5 Terminal Threat Hand-Off (TTH) Federate Operation**

The TTH federate located in the TCAC was started by executing the *handoff.sh* shell script. When the federate started, it prompted the operator for a run number. After a run number was entered, the federate prompted the operator to join the federation. When directed by the JADS test controller, the operator entered a ‘y’ at the prompt so that the federate joined the federation. The TTH federate waited until it received an *X\_File\_Spec* attribute update that contained the name of the script to be loaded. When the script name was received, the federate would display the name of the script that was being loaded. Upon completion of script loading, the federate

displayed “done.” The federate would then wait for an execution control attribute update with an execution control word that indicated start test execution.

During a federation execution, the TTH federate executed without operator intervention. The window in which the federate executed was monitored for error messages. The federate played its script until an execution control attribute update was received with an execution control word that indicated stop test execution. The TTH federate then resigned from the federation.

#### ***4.1.3.6 AFEWES Threats Federate Operation***

AFEWES threats federate consisted of software hosted on an SGI computer, facility unique systems and software for scenario status control and display, test management centers, and operator consoles. AFEWES controlled the test run execution and individual systems from a central facility linked internally by intercoms and external voice links to JADS and ACETEF. The test controller at JADS would advise the AFEWES controller by voice for federate and run starts and stops similar to the ACETEF federate. The AFEWES controller would coordinate internal execution actions with operators and advise JADS of current status.

#### ***4.1.3.7 Analysis Federate Operation***

The analysis federate provided an improved scenario viewer for observing northbound and southbound test runs and specific threat engagements. It showed the specific modes a threat site was using, and also showed missile flyouts as they occurred. Real-time displays of the ten EW Test MOPs, and the real-time values of jamming-to-signal ratio and tracking error were also provided for situation awareness of each threat engagement with the target aircraft. Data collection and storage for the analysis federate were nearly automatic and required little operator intervention once the run started.

## **4.2 Test Execution**

JADS achieved all test execution and data collection objectives established for Phase 2. The EW Test team executed the nine days of testing and exceeded the minimum goal of 62 runs per threat pair coupled with the several ADS excursion runs.

Phase 2 produced 363 total runs over nine days. Of those runs, 103 were aborted. The primary causes were federate failures (83) followed by procedure problems (10). Network problems accounted for two lost runs. Eight runs were tagged for further analysis since they showed high closed-loop latency ( $> 500$  milliseconds) which exceeded the design limit. These runs were allowed back into the valid test results when it was determined that the MOPs fit within the population as estimated by the good run measures of performance. On-site quality control at AFEWES indicated that all usable engagements provided data consistent with the HITL test accomplished in Phase 1.

This section reports on the test execution issues JADS experienced in Phase 2 using an ADS environment. Causes of problems and analysis of the impacts are provided if they were identified

during the conduct of the test. Much of the cause and effect examination of anomalies experienced was not accomplished until after the test during the detailed examination of all the run data by JADS analysts. Where problems are identified in this section without a specific resolution given, they are addressed in post-test analysis in Section 5.

Phase 2 exit criteria shown in Table 5 were all met. The successful test effort produced valuable data on EW and network performance. The ADS MOP results and evaluation are discussed in Section 5.

**Table 5. Exit Criteria**

<b>Objective #</b>	<b>Phase 2 Exit Criteria</b>
1-1	Complete at least 62 runs per threat pairs - twice ( 62x2 = 124 data samples)
1-2	Complete several ADS excursion runs
1-3	Establish ranges of DSM statistics for event data
1-4	Establish range of correlation coefficients for series observables
1-5	Quantify the effects of data latency on JADS/DSM test environment
1-6	Quantify the operating reliability and mean time between failure of the JADS network
1-7	Determine the connectivity performance of the JADS network

#### **4.2.1 Phase 2 Test Summaries**

Test execution of each run followed the same procedure. Through several risk reduction events, the test team determined that a stable test environment with the federates required a sequential, methodical joining process. Initially, the test began with the following sequence: RFENV federate, TTH federate, platform federate, TCF, ADRS, DSM federate, analysis federate and AFEWES federate. Although ADRS was not a federate, it provided the start and stop commands to the federates for each run. ADRS also sent a setup command to identify and load the appropriate scripts in the platform, RFENV, and TTH federates. Once the federates had completely joined and the AFEWES federate objects declared, the test controller began the run. Each run lasted approximately three-and-a-half minutes. By the fifth day of testing, the run turn-around time (from start of the last run to start of the subsequent run) was six-and-a-half minutes. Procedures had been streamlined. There were two major modifications implemented to further maximize the six-hour test day. The first modification was not to wait until each federate completely joined. Once the test controller observed a federate joining, the next federate was started. The other change involved joining while the aircraft script was being loaded in the platform federate. Normally the process was delayed until the platform federate finished loading the aircraft script before allowing the next federate to join. However, no degradation of the joining process was observed after establishing this change. If critical data elements were lost during execution of a run, federates crashed or connectivity was lost, simulators crashed or did not have a seemingly valid engagement, the run was terminated and not considered usable for analysis. Additionally, if an engagement was not scored as good by AFEWES (threat simulator

engagement), it was also counted as unusable for jammer MOP analysis but used for ADS assessment.

Phase 2 was conducted from 1 - 11 December 1998. During the first week (1-4 December), test runs were conducted with AFEWES SADS VIII and WEST X threat simulators manned during northbound and southbound runs. The WEST X was not activated during southbound runs to duplicate the Phase 1 engagement scenario. Scripts representing the other two threats (SADS III and SADS VI M) engagement activity against the jammer were generated by the RFENV federate. During the second week (7 - 11 December), test runs were made with AFEWES SADS III and SADS VI M threat simulators manned during northbound and southbound runs. Daily testing issues and results are summarized in Table 6 for all test runs. The number of total runs (341) does not include 22 ADS excursion runs and time synchronization runs. Problems encountered with specific components of the federation during testing are explained in the daily notes and subsections that follow.

**Table 6. Phase 2 Test Execution Summary**

	Total Runs	Maximum bandwidth used	Peak latency	Invalid runs	Effective rate	Invalid script	Comm or network error	TCAC software/hardware	AFEWES software/hardware	ACETEF software/hardware	Operator error
<b>DAY 1</b>	40	24%	262 ms	10	75%	1		6		2	1
<b>DAY 2</b>	42	38%	353 ms	10	76%			4	5		1
<b>DAY 3</b>	42	24%	12 sec	12	71%		1	4	1	4	2
<b>DAY 4</b>	35	56%	338 ms	6	83%		1	2		3	
<b>DAY 5</b>	17	23%	325 ms	6	65%			3	1	2	
<b>DAY 6</b>	36	23%	336 ms	12	67%		1	3	2	5	1
<b>DAY 7</b>	30	22%	376 ms	13	57%		1		7	3	2
<b>DAY 8</b>	53	23%	358 ms	12	77%			4	4	3	1
<b>DAY 9</b>	46	27%	346 ms	14	70%	1		5	3	5	

#### **4.2.1.1 Notes on Day 1**

For Phase 2, JADS analysts generated 744 scripts (each playback federate had a unique script for each run). Errors made in some scripts were recognized during the test runs and corrected.

#### **4.2.1.2 Notes on Day 3**

The maximum round-trip transmission latency measured for the reliable threat mode change and jammer response complex data types was 12 seconds. There were three other runs where the latency was higher than 4 seconds. In each of these cases, the DSM and/or AFEWES experienced abnormally high data dropout rates. This measurement throughout Phase 2 testing excluded AFEWES facility latency and the DSM SGI to PC latency. It included all RTI and long haul communications latency. Three unofficial runs were made to verify use of a network sniffer device at ACETEF with the DSM.

#### ***4.2.1.3 Notes on Day 5***

JADS completed the ADS excursion runs using all four threats (SADS VIII, SADS VI M, SADS III, and West X) active, then began normal Phase 2 test runs using the SADS III and SADS VI M active threats as scheduled. The ADS excursions were run first. The total runs completed were fewer than a typical test day for this reason.

#### ***4.2.1.4 Notes on Day 7***

Because of the high number of problems with both AFEWES threats, scripts were created with just one threat live and three from RFENV. This allowed completion of runs with only one live threat at AFEWES while the RFENV simulates the modes and codes of the other three.

#### ***4.2.1.5 Notes on Day 8***

JADS had a highly successful day of testing and completed a record number of ADS runs with the SADS III and SADS VI M engagements. Factors contributing to this included an additional hour of testing, AFEWES reliability was very high, ‘new’ federation problems were minimal, and the pace of test run execution and coordination among the three sites was exceptional.

### **4.2.2 Federate Summaries**

The total lost runs due to federate software crashes are summarized below. Federate software core dumps were the most frequent failure and caused the majority of lost runs during Phase 2 execution. The federates in the JADS TCAC were often impacted by failures on other federates, especially the TCF federate. If TCF crashed, the other federates would have to be restarted as well. Therefore, some federate totals (TTH, platform, and RFENV) are inclusive of TCF crashes.

**Table 7. Federate software core dumps**

Test Day	DSM	TCF	TTH	PLATFORM	RFENV
1	4	5	5	5	5
2	0	7	4	4	4
3	2	4	4	6	4
4	0	0	0	1	0
5	2	1	1	1	1
6	3	2	0	0	0
7	4	2	0	0	0
8	3	0	0	0	0
9	5	3	2	2	2
Federate totals	23	24	16	19	16

#### **4.2.2.1 Digital System Model (DSM) Federate**

The following problems and anomalies were experienced during the test with the DSM

**Missing Data to DSM.** The DSM operator at ACETEF reported numerous missing live entity state (LES) messages (aircraft TSPI) and threat performance (TP) messages (threat simulator modes) throughout several engagements on different test days. The DSM PC had six, external error displays at the bottom of its display (it also logged a number of other types of errors in one of its internal text files, *error.txt*). Three were for missing messages and three were for out-of-order messages. The messages it checked were platform live entity state; platform and AFEWES threat performance; and RFENV and AFEWES source mode change (SMC).

Because of the problem with the message sequence numbering, the AFEWES source mode change error displays were essentially useless. Error messaging was a unique DSM software capability initially used for debugging during DSM development.

The LES and TP error displays were somewhat useful, but the DSM PC code apparently didn't implement them properly since interruptions of platform data to ACETEF should have almost always produced equal numbers of lost LES and platform TP messages (that is what the DSM federate logs showed). However, the DSM often reported unequal numbers in those cases.

A final DSM problem was that an attempted recompile and rebuild to go back to the original code did not produce a DSM user interface executable file that was the same size as the original one GTRI built and used. The previous week's executable file was ultimately restored and used for the rest of the test. JADS did not know exactly what source code GTRI used to create the executable hosted on the DSM PC.

**DSM Federate Core Dumps.** DSM federate core dumps on the SGI O2 occurred at a high rate during testing. The cause was not determined; however, subsequent reruns of the scripts in use when the core dumps happened were always successful. There appeared to be two kinds of DSM and TCF core dumps: those that occurred because the DSM PC (or an ADRS PC) was started too soon in the federation startup process, and those that occurred after the PC had established a TCP connection with the federate.

**DSM Threat Mode Limit.** Early in the second week of testing, the test team noted that the age-out times were not consistent with what had been expected. This anomaly was observed predominately with the SADS III and occasionally with the SADS VI M. A preliminary investigation of the DSM code was conducted to ensure that code changes implemented during the ADS excursion runs had not corrupted the software. The change was backed out by rebuilding the executable with the original source code file as described above (see missing data to DSM). When that failed to resolve the problem, the previous week's backup copy of the DSM executable was restored. This also failed to eliminate the problem.

When the relevant PC code was examined, a hard coded limit of the threat mode changes allowed into the DSM was noted. This buffer permitted only 112 messages of this type. When the value was exceeded, the additional mode change messages were essentially ignored. Analysts reviewed the good runs with the SADS VIII and WEST X active. These runs contained about 50 mode changes received at the DSM logger. The number of modes received during the unexpected age-out times observed was also checked. The values were consistently above 112. GTRI software engineers verified the 112 limit was due to the buffer. The effect of this anomaly on the SUT MOPs was handled during post-test analysis and a work-around for most of the age-out problems was developed during post-test analysis. The incorrect jammer age-out times for the threats continued throughout the remainder of Phase 2. This was not impacting the goodness of the runs, however, since it occurred at the end of the DSM threat engagement for a specific threat. JADS EW Test MOPs do not include age outs. As long as jamming started against the threat, the engagement proceeded normally but jamming never turned off.

**SGI Problem.** ACETEF encountered a computer problem when the federate (SGI O2 computer) stopped creating output files. Rebooting the computer solved the problem and no runs were aborted.

**DSM Logic Flaw.** During a single engagement on day one (run 7), the response from the DSM did not match what was expected. The DSM appeared to incorrectly engage the SADS VIII after the SADS VIII took an unusually long time acquiring the aircraft. The SADS VIII jamming was prematurely terminated. Also observed in the engagement were a higher than normal number of threat mode changes and jammer responses. GTRI attempted to solve the problem by changing the DSM PC software. The revised DSM software was installed after the first week of testing, however, it did not resolve the problem. After accomplishing several ADS excursion runs, the DSM code from the previous week was used. All runs for record completed the second week used the original (week one) DSM code. Reruns were completed successfully.

#### **4.2.2.2 Test Control Federate (TCF) and ADRS Operation**

The following problems and anomalies were experienced during the test with the TCF and the ADRS computers.

**Script Loading Failures and TCF Core Dumps.** On several occasions, the TCF was unable to load the platform script after receiving the ADRS setup command. This was usually caused by starting ADRS too soon or selecting an ADRS menu option too soon. When ADRS starts, it must set its time to match the federate time. This can take several seconds. If you select an ADRS menu option while it is synching its time, it will cause a TCF federate crash. Sometimes following an aborted run, the next run was aborted for the same problem. On the third attempt, the run was successfully executed. The test team examined the federate files to pinpoint which system failed to carry out its assigned task. This problem occurred less frequently as the test progressed.

**ADRS Crashes.** Frequently at least one of the ADRS computers crashed during the test runs. When this was observed during integration, JADS added more ADRS computers to the federate. This response was driven by perceived risk compared to other problems that were being worked at the same time. Crashes usually resulted in leaving an open TCP connection on the TCF federate computer. A TCP connection is a one-to-one communications link defined by the IP addresses of the two hosts and the two TCP port numbers of the sender and receiver. This problem arises from the basic design of how TCP handles intermittent data, lost communications, etc. Overall, the next run was delayed by several minutes while the connection between the two machines timed-out and TCF was restarted.

#### **4.2.2.3 Platform Federate**

The following problems and anomalies were experienced during the test with the platform federate.

**Core Dumps.** The federates residing in the TCAC (TCF, platform, TTH and RFENV) had multiple core dumps. The cause was not determined; however, subsequent reruns of the scripts were successful. One failure happened with the platform federate at the end of the run. Platform also crashed before the ADRS computer sent the start command because it failed to load the aircraft script. Again, this was observed during the integration. At the time, the GTRI software developer forwarded a theory to explain the problem, but JADS was unable to verify it with the RTI support desk.

**Script Errors.** Several minor platform script errors were noted. All the errors were fixed and new scripts generated. On day three of the test, the platform federate crashed before the start command was sent by the ADRS computers, and the aircraft script failed to load.

#### **4.2.2.4 Radio Frequency Environment (RFENV) Federate**

The following problems and anomalies were experienced during the test with the RFENV federate.

**Core Dump.** The federates residing in the TCAC (TCF, platform, TTH and RFENV) had multiple core dumps.

**Bad RFENV Script Loads.** RFENV federate failed to load the scripts on two different runs. New REFENV scripts were transferred to the RFENV federate and no other problems occurred. A small number of suspected script anomalies were observed, but our initial assessment determined no impact on the data collected. All the anomalies were apparently early activations of the SADS III on southbound runs.

#### **4.2.2.5 Terminal Threat Hand-Off (TTH) Federate**

The only problems experienced during the test with the TTH federate were that the federates residing in the TCAC (TCF, platform, TTH and RFENV) had multiple core dumps. The cause was not determined; however, subsequent reruns of the scripts were successful.

#### **4.2.2.6 AFEWES Threats Federate**

The following problems and anomalies were experienced during the test with the AFEWES federate.

**No Emitter Mode Verification Instrumentation (EMVI).** JADS tried to get an EMVI from the Eglin AFB test range to time tag each pulse from JETS. These data could then be used to calculate the AFEWES internal latency from TAMS to JETS. The EMVI shipped from Elgin AFB to AFEWES had numerous problems and broken components. Consequently, the EMVI was not available to support Phase 2, and the appropriate instrumentation was not available to collect latency data between the TAMS and JETS.

**AFEWES Outages.** AFEWES had a problem initializing the TMC. This problem only delayed the test start time; no runs were aborted. AFEWES also reported a problem with an RF head—one run was aborted. The problem was resolved by the time testing resumed. At the beginning of day seven, AFEWES had a problem with the optical disk in a TMC. This problem only delayed the start time; no runs were aborted.

**SADS III.** The DSM operator reported a problem during a SADS III engagement run. A subsequent investigation revealed large tracking errors and this run was aborted. SADS III experienced several additional problems throughout the entire test day. On the tenth day of testing, the system crashed twice (runs 313 and 316).

**SADS VI M.** Test observers noted two runs with unusual engagements; higher than normal miss distances were observed. The suspected cause was late activation of the jammer technique at AFEWES because of high latency associated with the SUT jammer technique message to activate the JETS. Subsequent analysis did not show any excessive latency external to AFEWES. ADS did not impact these miss distances. Other problems were traced back to the SADS VI M computer system. The SADS VI M exhibited continuing faults throughout one day. On two occasions the track modes were not operating correctly, and then testing had to be shut down until a RF head power source was replaced. AFEWES traced a single problem on run 303 back to the SADS VI M computer system. Reliability improved as the test progressed.

**WEST X Off Commands.** At the end of two different runs JETS at AFEWES did not receive the West X "off" mode. (JETS supplied the RF energy to the active threats at AFEWES.) After reviewing the log files (runs 36 and 37), analysts located the appropriate "off" mode from the DSM commanding the JETS to stop radiating; therefore, the data were valid. The data contained in the logger file outline a "proper" engagement.

**West X Computer Crashes.** During two successive runs the Honeywell computer crashed. Southbound engagements were then run until the system was operational again. The West X participated with the SADS VIII only in the northbound engagements. Once the system was restored, normal engagements were resumed according to the execution matrix. This matrix contained both northbound and southbound runs.

#### *4.2.2.7 Analysis Federate*

During excursion runs, the analysis federate was unable to handle all four threat simulators producing live data at the same time. A buffer in the system would fill about three-quarters full, and the application would core dump and close the Vision 21 application (federate software) near the end of each of those all four live threat runs. Some MOP calculation algorithms were not completed when the test was executed, but this did not impact real-time analysis capability because the incomplete MOPs were backups to the ADRS displays. The analysis federate also loaded quite slowly because of the processing power of the Sun workstation. This caused the analysis federate to almost always be the last federate to join the federation and occasionally delayed runs by 30-60 seconds. The full potential of the analysis federate was never realized. Graphical representations of flight profile and missile flyout were useful. Numeric measures beyond miss distance were seldom used during test execution. These numbers were more difficult to interpret at a glance than the graphical ADRS displays.

#### **4.2.3 Runtime Infrastructure (RTI)**

Test observers noted several RTI error messages during Phase 2. A DMSO RTI representative was viewing the test and passed the error messages to the RTI developers for investigation. The developers confirmed the messages came from the RTI but did not pursue investigation much further as later releases of the RTI were available to HLA users.

#### **4.2.4 Wide Area Network**

The following problems and anomalies were experienced with the network.

**T-1 Line Drops.** AFEWES lost two seconds of TSPI in the middle of run 52. Simultaneously, the KIV-7HS crypto device signaled an alarm. It was located in a communication rack several feet behind the AFEWES federate operator. After N&E investigated the problem, it was determined that the T-1 line between JADS and AFEWES had lost synchronization and the cryptos were resynchronizing. AFEWES observed no adverse effect because of TSPI dropouts. On day seven, the link between ACETEF and JADS lost crypto synchronization during a run. The run was aborted and the cryptos automatically resynchronized within a minute.

**ACETEF Latency Spikes.** Due to sporadic latency spikes at ACETEF, a network sniffer was connected to collect the times in which data were received and transmitted from the DSM SGI to the DSM PC. The sniffer data files were expected to aid our efforts in characterizing the DSM latency.

**User Datagram Protocol (UDP) Data Losses.** On day two during run number 53, approximately 1500 platform script data packets sent unreliable were lost. The cause of this large number of data dropouts was not determined. Sporadic losses of UDP data sent across the network continued regularly. N&E performed several UDP raw network tests (without the RTI) during breaks. They sent more than 72,000 packets and lost 45. When dropouts occurred, the network pings and the IDNX historical database provided verification that the T-1 links were not down. JADS N&E continued pursuing this problem. On day eight of the test, JADS N&E noted seeing only minimal dropouts of packets through the routers. Considering the number of packets sent, the number of dropped packets was within range. On day nine multiple occurrences of lost packets of UDP data sent across the network were reported by AFEWES and ACETEF (runs 266, 301, and 311). N&E noted seeing only minimal dropouts of packets through the routers. When dropouts occurred, the network pings and the IDNX historical database continued to provide verification that the T-1 links were not down. The routers were periodically rebooted during the day based on lessons learned from other JADS tests. On the last day of testing, similar to day eight, multiple occurrences of lost packets of UDP (unreliable) data were reported by AFEWES and ACETEF (runs 324, 356, 347 and 357). N&E continued to note only minimal dropouts of packets through the routers.

#### **4.2.5 Data Collection**

There were two distinct problems with data collection. The first was the use of ADRS in collecting MOP data to be summarized upon test completion. The ADRS MOP algorithms were not completed by the test start, so only partial MOP data were available for examination at the end of each test day using ADRS. This never negatively impacted the test because all troubleshooting efforts were done using the real-time displays already available in ADRS. The impact of this lapse in data collection occurred post-test when the analysis process began, and all runs had to be reduced in ADRS once the final version was delivered.

The other problem was due to the unavailability of the appropriate instrumentation to collect latency data between the JETS and TAMS. This measurement could have better characterized the network.

#### **4.2.6 Test Control Outages**

During day three, a TCAC circuit breaker tripped and resulted in the loss of power to all computer monitors and the ADRS 2 and 3 processors. The breaker was reset and testing resumed. The federate computers were not affected because they are connected to a backup power source. However, due to the increased number of computers and large monitors in the TCAC (all receiving power through a single breaker), N&E had another breaker added for the TCAC before Phase 3. On day four, for approximately one hour, ACETEF experienced a voice communication outage. Additional runs using an unclassified phone line were conducted until the problem was resolved. The communication link was restored after reseating the voice card in the IDNX at ACETEF. During this process, however, the IDNX was powered off. The ACETEF network technicians did not know that the voice card could be removed and replaced while the IDNX power is still on. The IDNX software is stored on memory boards in a compressed format. It took about 15 minutes to reboot the operating system and bring the node on line. The test director ensured proper coordination among JADS N&E and the distant nodes for corrective actions on network-related problems.

#### **4.2.7 Test Execution Lessons Learned**

During daily testing, a run matrix was used to determine which runs were executed, what procedures were used to start the federates, models and simulators, and to initiate the run, stop the run and shut down of the federates. The work done during preliminary testing (e.g., FAT, FIT) provided a repeatable methodology for orderly federation operation that was used for the formal test. Nonetheless, problems were frequently encountered, errors were made, and unanticipated issues arose. The areas critical to performing distributed testing that yielded valuable lessons learned are discussed below. Lessons learned or solutions are provided based upon JADS test requirements.

##### **4.2.7.1 Software/Hardware Reliability Issues**

ADRS equipment crashes and reboots frequently disrupted testing and slowed the rate of testing. The problem was moderated during Phase 2 by adopting new procedures like rebooting each time a computer was idle. SGI O2 to PC interface software developed for our federates (called the JADS communicator) would leave a communications socket allocated after ADRS crashed, so additional time was lost waiting for the socket to reset afterwards. Procedural speed for starting ADRS contributed to the problem. It was very important that the computer be started in a specific sequence in the TCAC. The action adopted by JADS was to analyze the run logs for reliability problems to determine where changes in processes and communication could improve Phase 3 operations. A memory leak problem was detected in the ADRS computer that resulted in

many software crashes caused when the system would run out of available memory. This was solved by the frequent reboots. It was also marked for correction before Phase 3.

#### **4.2.7.2 Test Rehearsal**

The FAT and FIT series of federation tests were invaluable for establishing Phase 2 procedures within the TCAC and with AFEWES and ACETEF. However, personnel were added or changed locations for test execution which impacted test rehearsal learning. The action adopted by JADS was to plan appropriate test rehearsals and comprehensive integration tests for Phase 3.

#### **4.2.7.3 DSM Performance**

Some problems existed in the robustness of the DSM software that resulted in reliability problems with the DSM and its federate. Although a hindrance, the DSM performance was adequate to collect the needed data. Better software design, quality assurance, and testing would have uncovered the reliability, logic, and buffer limit issues seen in the test execution.

#### **4.2.7.4 Site Manning/Workload During Test Execution**

The number of computers, intricate execution procedures, and high number of test events performed sequentially created a very workload intensive environment at the TCAC and other locations during testing periods. Manning requirements at the TCAC, AFEWES, and ACETEF involved 14 dedicated JADS personnel during the two-week test period. Site manning (3 persons) at AFEWES was insufficient. It was determined that 1 additional person would be required for rotation among JETS, TAMS and the simulator stations. JADS reviewed and updated the site manning matrix for Phase 3.

#### **4.2.7.5 Tools and Procedures for Real-Time Analysis of Run Goodness**

During test runs, the TCAC test controller was highly dependent on ADRS for federation and scenario status monitoring. Analysts were highly dependent on the ADRS emitter state history display for monitoring jammer/threat engagement details. JADS found the analysis federate scenario visualizer to be a solid capability. While it provided an extra set of graphical displays of the unfolding engagement, it also provided real-time feedback of the EW Test MOPs. Anomalies in miss distance and response times could be instantly assessed which was an added capability separate from ADRS. The analysis federate could not take the place of an ADRS machine for Phase 3, but it could support troubleshooting of anomalies seen during the runs. Without the analysis federate, problems seen could be at AFEWES, ACETEF or in one of the many federates run at JADS. The analysis federate aided in identifying the source of the problem.

If presented with extremely limited time and manning, the analysis federate could be eliminated with only a small impact to test execution. It was not critical to the function of the test but did provide an extra source of examination of the run execution. The greatest benefit of the analysis federate was the real-time assessment of the EW Test MOPs and the integration of the aircraft

profile with the threat mode status. If tasks needed to be combined, the analysis federate would need to be updated to assume the responsibilities of the second ADRS machine.

#### ***4.2.7.6 Voice Communications.***

JADS voice links were conducted using conference calls with open lines to AFEWES and ACETEF. This capability continued to evolve as command and control requirements evolved. JADS used head-mounted earphone/microphone equipment and experienced numerous problems with hearing and being heard across the network. Most problems were alleviated with equipment familiarity and experience. Batteries had to be replaced frequently. The FAT and FIT demonstrated shortfalls in the voice communications that required more equipment at each facility. As the number of instruments increased, testers became very busy coordinating status, communicating test information, and controlling run execution. Consequently, message transmission length had to be minimized; external background conversations avoided; and test problem troubleshooting had to be done via a separate line. ACETEF had some telephone instrument problems. JADS will utilize Phase 2 conference call initialization methods during Phase 3 and include more formalized discussion procedures and protocols.

#### ***4.2.7.7 Network***

TSPI data losses among platform federate, the DSM, AFEWES, and TCF occurred frequently during federation integration. JADS and DMSO investigated this problem. A work-around solution was found. This solution was a fixed join process for federates prior to each test run. After the solution was implemented, data losses among the DSM, AFEWES, and the platform federate were still observed. These losses manifested themselves in the apparent hovering aircraft observed in the FIT. It was not clear what caused the data loss, but dead reckoning aircraft position provided an acceptable solution. This data loss coupled with the dead reckoning implementation at AFEWES was the suspected cause of an extremely large miss distance value on one missile shot. RTI bundling of federate data for transmission made troubleshooting data flow and transmission problems more difficult. Our tools assessed hardware performance only.

Instrumentation for federation performance evaluation used in Phase 2 was inadequate. It lacked the ability to examine data passed between RTI instances. Best effort data could be dropped by the network without notification or without any faults reported by the hardware. Phase 3 network instrumentation must be expanded to include network sniffer to monitor network traffic between the sites.

#### ***4.2.7.8 Test Control Procedures***

JADS refined voice protocols for acknowledging readiness among sites and starting/stopping runs for Phase 3. Further review of link health status confirmation procedures and network health check impacts was needed. JADS improved situational awareness for network health and readiness across sites and formalized the procedures as necessary.

#### **4.2.7.9 Software Changes**

Configuration changes on tools (analysis federate, ADRS display, DSM joining process, AFEWES dead reckoning algorithms) could have severe impacts. Configuration changes, even seemingly trivial ones, must be coordinated at all levels. JADS stressed configuration management procedures for Phase 3 and will enforce their use.

#### **4.2.7.10 Latency and Time Synchronization**

JADS was highly dependent upon time synchronization of all federate computers and software. However, any requirement for synchronization requires the ability to verify the requirement is being met. For example, if one millisecond synchronization accuracy is required, then a capability to measure time between two computers at one millisecond precision is necessary. However, software tools were not available to measure accuracy at that level. In fact, testing time synchronization across the federation was more art than science. Even with time synchronization and the time cards implemented in all computers, instances were noted where time synchronization “slipped” affecting latency measurements. A few occurrences of time synchronization problems across the federation were observed requiring JADS to research particular runs after daily testing. JADS consistently followed documented time synchronization procedures and hardware settings. Site support personnel were relied upon to implement procedures and verify settings daily.

#### **4.2.7.11 Run Speed/Time Between Runs**

Operator boredom with the repetitiveness of test runs at AFEWES may have contributed to afternoon run differences. JADS attempted to minimize turn-around time between runs. The time between runs from start time to start time was usually between 6 and 8 minutes.

#### **4.2.7.12 RTI Heartbeat**

This health check was inadequate for monitoring JADS federation status for several reasons. First, its time scale, which was about 20 seconds before any indication of a problem, was too long for a real-time federation. Second, because it sent its messages via TCP/IP, this system could not detect a problem for federate messages sent via RTI best effort, i.e., UDP/IP-based protocol, unless the underlying cause of the problem affected both of those protocols. And third, the RTI did not time stamp and log these messages, so they were only available in real time. JADS did not use this as a primary indication of federation health so no changes were required. Future federations should investigate RTI tuning features (e.g., runtime infrastructure initialization data [RID] file parameters) or other RTI management features (e.g., management object model calls) if the federation doesn't implement its own health monitors like JADS did.

#### **4.2.7.13 Federate Link Health Check (LHC)**

The JADS link health check scheme provided reasonable insight into federation health once it was understood. Analysis showed that there was a high correlation between the loss of LHC messages and most, but not all, events that involved the loss of other federate messages sent best effort and/or the delay of messages sent via RTI reliable, TCP/IP-based communications protocol. Due to its 1 hertz message frequency, the LHC system sometimes missed best effort data loss events lasting less than 1 second, but those events apparently did not cause any simulation problems. Since the LHC system sent its messages via the best effort protocol, it could also not detect short-duration problems that affected only the TCP/IP connection used for reliable protocol between two federates.

Perhaps, the most interesting result from the post-test analysis of the LHC messages was that the LHC system detected selective, one-way best effort data losses between federates that may be a symptom of problem(s) with the RTI's use of IP multicast groups. For the runs during which these problems were observed, the losses were selective because LHC messages (and usually other federate messages sent best effort as well) were lost between one or more federates at JADS and the federate at another test node, but not between the remaining JADS federates and that remote federate. The losses were one-way because the LHC messages between the federates experiencing the problem were lost in only one direction. Typically, during such events, there was no delay in the flow of reliable messages between those federates, if any reliable traffic was present. It was difficult to understand how network or network hardware problems could produce such selective, one-way data losses. While LHC as implemented has limitations, it will be sufficient for JADS in Phase 3. Future federations should consider the limitations noted above if they choose to pursue a similar health monitor scheme for their federation.



## **5.0 Data Analysis**

The test collected two classes of performance data: jammer MOP and ADS measures. The detailed jammer MOP results will be covered in a separate classified report containing both the Phase 2 and Phase 3 data analysis. The results of jammer MOP correlation across the EW Test phases are presented in Section 6.0. Since the JADS test program was designed to evaluate the utility of ADS for EW T&E, the analysis of jammer performance was not the primary focus of the test. Instead, the study of the impacts of ADS on the jammer MOPs provided a method to assess the utility of ADS within the JADS EW Test environment. This section addresses the detailed analysis of the ADS measures.

### **5.1 ADS Measures**

This section describes the relationship between the EW Test objectives for Phase 2 and the ADS measures that ultimately support JADS-level objectives and issues. Table 8 shows how it was anticipated that EW Test activities would provide information for addressing JADS objectives from an EW perspective by showing corresponding EW and JADS objectives. Table 9 lists the individual JADS-level ADS measures evaluated during Phase 2. As stated previously, the Phase 2 ADS measure data will be used for subsequent comparison with data obtained from Phase 1 and Phase 3.

**Table 8. JADS and EW SPJ Test Objectives Correspondence Matrix**

SPJ Obj #	Self-Protection Jammer (SPJ) Test Objectives	Expected JADS Objectives Supported by SPJ test (per May 96 EW Test APA)	JADS-Level ADS Measures Supported*
<b>1</b>	<b>Measure SUT performance data in each environment</b>	<b>Subobj 1-2-2:</b> Assess ADS capability to support live T&E planning and test rehearsal	
1-1	OAR (Baseline)		
1-2	DSM		1-2-2-2, 1-2-2-3, 1-2-2-4
1-3	ISTF		
<b>2</b>	<b>Establish repeatability of OAR and ADS test results</b>	<b>Subobj 1-2-2:</b> Assess ADS capability to support live T&E planning and test rehearsal	
2-1	OAR (Baseline)		
2-2	DSM		1-2-2-2, 1-2-2-3, 1-2-2-4
2-3	ISTF		
<b>3</b>	<b>Correlate data between environments</b>	<b>Subobj 1-2-2:</b> Assess ADS capability to support live T&E planning and test rehearsal	
3-1	OAR-DSM duplicated threats		1-2-2-2, 1-2-2-3, 1-2-2-4
3-2	OAR- ISTF duplicated threats		
3-3	OAR-HITL duplicated threats		
<b>4</b>	<b>Quantify the effects of ADS-induced errors</b>	<b>Obj 1-1:</b> Assess validity of data from tests utilizing ADS <b>Subobj 1-2-2:</b> Assess ADS capability to support live T&E planning and test rehearsal <b>Subobj 2-1-2:</b> Assess network and communication performance constraints and concerns	
4-1	Latency on ADS test results		2-1-2-4
4-2	Effects on human perception		1-1-0-3, 1-1-0-4
4-3	Others		
<b>5</b>	<b>Measure ADS network performance</b>	<b>Subobj 2-1-2:</b> Assess network and communication performance constraints and concerns <b>Subobj 2-1-3:</b> Assess the impact of ADS reliability, availability, and maintainability	2-1-2-1, 2-1-2-2, 2-1-2-3, 2-1-3-3
<b>6</b>	<b>Measure ADS reliability</b>	<b>Subobj 2-1-2:</b> Assess network and communication performance constraints and concerns <b>Subobj 2-1-3:</b> Assess the impact of ADS reliability, availability, and maintainability	2-1-2-2, 2-1-2-3, 2-1-3-1, 2-1-3-2, 2-1-3-3

\* JADS-level ADS measures not listed are assessed indirectly after test completion

**Table 9. JADS Measures Evaluated During Phase 2**

JADS EW ADS Measure	Title
1-1-0-3	Degree to which test participants were able to distinguish between ADS (virtual or constructive) versus live assets
1-1-0-4	Degree to which test actions were impacted due to the ability to distinguish between ADS and live (non-ADS) targets
1-2-2-2	Degree to which test control procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of test control procedures
1-2-2-3	Degree to which data management procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data management procedures and tools
1-2-2-4	Degree to which data reduction and analysis procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data reduction and analysis procedures and tools
1-2-3-3	Degree to which ADS can increase test times, events, etc.
2-1-1-1	Degree to which live, virtual, and constructive entities exist, can be instrumented, and can be readied for a test
2-1-2-1	Average and peak throughput available for each link
2-1-2-2	Percentage of complex data types received out of order by a federate
2-1-2-3	Percentage of total complex data types subscribed to by a federate that was received by the federate
2-1-2-4	Average and peak data latency
2-1-3-1	Degree to which test events (trials) were affected by ADS components (failure or otherwise) exclusive of network problems
2-1-3-2	Degree to which test events (trials) were affected by network problems (failure or otherwise)
2-1-3-3	Degree to which test events (trials) were affected by personnel problems
2-2-1-4	Ease with which data can be retrieved, post-trial, from a given node
2-2-2-1	Degree to which test managers can control the configurations of ADS participants, the ADS environment data, and ADS networks
2-3-2-3	Degree to which protocols, processes, and procedures are needed to enable effective centralized test control
2-3-2-4	Degree to which real-time analysis systems support test safety and other test control requirements

### **5.1.1 Measure 1-1-0-3. Degree to which test participants were able to distinguish between ADS (virtual or constructive) versus live (non-ADS) assets.**

**Intent.** The intent of this measure was to determine if workstation operators at AFEWES could distinguish between ADS-linked assets and non-ADS assets (SPJ installed at AFEWES) and, if so, to what extent.

**Data Collection and Analysis Approach.** Interviews were conducted with AFEWES test participants concerning their perceptions and actions. Interview questions focused on procedural and technical differences between the EW ADS test and other non-ADS test events with which participants had experience. AFEWES operators were asked to describe the impacts of any unusual procedures or unrealistic behaviors on their ability to perform test operations. JADS analysts reviewed and summarized recorded remarks.

**Data Sources.** Eight interviews were conducted. Interviewees included the AFEWES federate controller, the TAMS operator, and the SADS III, SADS VI M, SADS VIII, and WEST X threat system operators.

**Results.** AFEWES operators discussed several procedural differences between ADS and non-ADS testing noting in particular the increased need for strict equipment configuration control and checklist adherence while setting up for ADS runs. They also noted that more threat simulations were run simultaneously during ADS testing, resulting in the need for increased layers of command and communication. Technical differences included some software interface problems and slower initialization of runs. SADS III and SADS VIII operators discussed seeing some unusual target behaviors such as “jumping” and “jittering” during ADS runs, especially for high-speed targets. The SADS VI M and WEST X operators indicated that target behaviors looked normal, except for occasions when the target stopped entirely. The TAMS operator noted that, in general, all ADS target data looked more coarse than non-ADS and jumped to their true position upon run start. (JADS on-site observers looked for these effects during the test execution and were instructed to note any problems that occurred during engagements. Only one run was lost because of jump when the flyout model failed after the target transitioned from dead reckoned position to actual position. JADS observers noted that several scripts showed unrealistic behavior immediately at start-up. These problems went away before the threat engagements began).

**Conclusions/Recommendations.** While AFEWES operators noted procedural and technical differences that enabled them to distinguish ADS testing from non-ADS testing, there did not appear to be any major issues or problems stemming from the differences. High-speed maneuvering targets would likely be more impacted by the data losses and dead reckoning that caused the jumps in the JADS architecture.

#### **5.1.2 Measure 1-1-0-4. Degree to which test actions were impacted due to the ability to distinguish between ADS (virtual or constructive) and live (non-ADS) assets.**

**Intent.** The intent of this measure was to determine if being able to distinguish between ADS-linked assets and non-ADS assets impacted AFEWES workstation operators’ actions, particularly actions that affected SUT MOPs.

**Data Collection and Analysis Approach.** Interviews were conducted with AFEWES test participants concerning their perceptions and actions. Interview questions focused on procedural and technical differences between the EW ADS test and other non-ADS test events with which participants had experience. AFEWES operators were asked to describe the impacts of any

unusual procedures or unrealistic behaviors on their ability to perform test operations. JADS analysts reviewed and summarized recorded remarks.

**Data Sources.** Eight interviews were conducted. Interviewees included the AFEWES federate controller, the TAMS operator, and the SADS III, SADS VI M, SADS VIII, and WEST X threat system operators.

**Results.** AFEWES operators indicated no negative impacts of ADS test procedural differences on their ability to perform test operations. If any impact was noted, it was that running multiple threat system simulations simultaneously and not having to create scenario tapes made the TAMS operator's job easier. On the other hand, some threat operators identified impacts of ADS technical differences on their ability to perform. The SADS VI M operator occasionally encountered SUT data losses when target data stopped prematurely toward the end of the run. The SADS VIII operators had difficulty tracking high-speed targets when they jumped out of gates. The TAMS operator noted having to accommodate position discontinuities via system software and overlook jumpy target data in analysis. AFEWES operators also felt that the slow ADS run initialization process and loss of aborted run data resulted in less SUT data being collected than could be with non-ADS test methods.

**Conclusions/Recommendations.** Some AFEWES threat system operators noted technical differences between ADS testing and non-ADS testing that impacted their ability to track targets and collect SUT data. Phase 2 SUT MOP analysis was performed with this knowledge; JADS analysts noted AFEWES operator log comments and carefully researched any potential SUT data anomalies. Another significant issue was that of lost runs and time between runs. ADS architectures require coordination across several facilities which can increase the time between runs and affect operator performance. Future ADS testers need to be aware of the potential for impacts.

#### **5.1.3 Measure 1-2-2-2. Degree to which test control procedures are impacted by ADS and how ADS can impact the pretest development and rehearsal of test control procedures.**

**Intent.** This measure was intended to evaluate the impact of ADS on test control procedures including development and rehearsal by comparing test control procedures for ADS versus non-ADS testing.

**Data Collection and Analysis Approach.** Interviews were conducted with test controllers and test executors to ascertain their perceptions about ADS test control procedures. Interview questions required an assessment of the quality and complexity of ADS test control procedures, as well as the potential differences between ADS versus non-ADS test control procedure development and rehearsal. JADS analysts reviewed and summarized recorded remarks.

**Data Sources.** Seven interviews were conducted. Interviewees included the two JADS test controllers positioned in the TCAC, the two JADS test executors positioned at AFEWES and ACETEF, and three test station operators positioned in the TCAC.

**Results.** Interviewees described test control procedures developed and implemented for Phase 2 (ADS) including those for pretest coordination, voice communications initialization, time synchronization and network verification tests, federation joining, and test event start-up. Although these procedures were rehearsed prior to testing, during integration and acceptance testing events and followed by using strict checklists, consensus was that the procedures were sometimes not sufficiently rigid or detailed to provide a desired level of control. At times, flexibility was required as procedures needed to be altered during actual test events to facilitate better communications between and within sites. Some test control procedures, such as methods for identifying, communicating, and controlling test event faults and anomalies, were developed more informally during rehearsal and test events. As test control procedures improved, there were fewer trial losses due to miscommunication. Test processes became more efficient, resulting in higher data quality and quantity.

**Conclusions/Recommendations.** In general, interviewees identified the need for flexibility in developing and rehearsing test control procedures as the main difference between ADS and non-ADS testing. Initial checklists provided a framework that was improved as it became clear during rehearsal events who had the best vantage point for coordinating the distributed team. Voice protocols were also modified to avoid miscommunication while keeping all key players informed and aware. The availability of a single voice channel for test control was the key to resolving issues on the fly. More up-front planning and rehearsal would have improved efficiency.

#### **5.1.4 Measure 1-2-2-3. Degree to which data management procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data management procedures and tools.**

**Intent.** This measure was intended to evaluate the impact of ADS on data management procedures and tools including development and rehearsal by comparing data management procedures and tools for ADS versus non-ADS testing.

**Data Collection and Analysis Approach.** Interviews were conducted with Phase 2 data managers and analysts to ascertain their perceptions about ADS data management procedures and tools. Interview questions required an assessment of the quality and complexity of ADS data management procedures and tools, as well as the potential differences between ADS and non-ADS data management procedures and tools development and rehearsal. JADS analysts reviewed and summarized recorded remarks.

**Data Sources.** Eight interviews were conducted. Interviewees included the TCAC data manager, the AFEWES and ACETEF data management representatives, and five JADS data analysts.

**Results.** Interviewees described data management procedures developed and implemented for Phase 2, as well as tools that were developed or acquired for managing test data. These included the federate data loggers, visualization tool data archiving software, and various UNIX and PC-based file transfer protocols. The consensus was that the fairly informal procedures used for consolidating, transferring, and storing data were sufficient to meet the needs of the close-knit

analyst team. Procedures were rehearsed during and after integration and acceptance testing events. The detailed coordination across distributed sites needed for developing a sound data management plan and the electronic transfer, storage, and accurate retrieval of multiple different types of data from distributed federates were cited as major distinctions between ADS and non-ADS testing.

**Conclusions/Recommendations.** Regardless of the type of testing that is conducted, i.e., ADS versus non-ADS, the key to data management is developing and rehearsing a plan that sufficiently addresses the consolidation, transfer, storage, and retrieval of the data needed for sound SUT analysis. ADS only adds the potential that data will be collected at more than one site and by more than one responsible organization. This pushes more responsibility for planning onto the tester. Tools can be developed or acquired to meet particular data management needs, whether for small or large amounts of data, and whether generated by a single or multiple facilities. Careful planning and rehearsal ensure data management processes run smoothly when test time is limited and personnel are busy with other issues.

#### **5.1.5 Measure 1-2-2-4. Degree to which data reduction and analysis procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data reduction and analysis procedures and tools.**

**Intent.** This measure is intended to evaluate the impact of ADS on data reduction and analysis procedures and tools including development and rehearsal by comparing data reduction and analysis procedures and tools for ADS versus non-ADS testing.

**Data Collection and Analysis Approach.** Interviews were conducted with Phase 2 data analysts to ascertain their perceptions about ADS data reduction and analysis procedures and tools. Interview questions required an assessment of the quality and complexity of ADS data reduction and analysis procedures and tools, as well as the potential differences between ADS and non-ADS data reduction and analysis procedures and tools development and rehearsals. JADS analysts reviewed and summarized recorded remarks.

**Data Sources.** Seven interviews were conducted. Interviewees included the TCAC data manager, the ACETEF data management representative, and five JADS data analysts.

**Results.** Interviewees described data reduction and analysis procedures developed and implemented for Phase 2, as well as tools developed or acquired for reducing and analyzing test data. These included the federate data loggers, visualization tools and SUT data reduction software (e.g., ADRS, analysis federate), various C++ log file summary and comparison software utilities, and various PC-based statistics and analysis packages (e.g., Excel<sup>®</sup>). Rehearsal of data reduction and analysis procedures took place during integration and acceptance testing events giving analysts an opportunity to become familiar with the tools and analysis procedures, as well as ensuring that planned tools could handle the amount and types of data collected. Both informal and formal training sessions were held for analysts to “dry run” the tools. Interview respondents identified one drawback for rehearsal of data reduction and analysis tools for ADS; it may not be possible to rehearse automated reduction system use without participation from personnel at all

involved sites. On the other hand, ADS testing does provide the benefit over non-ADS testing of being able to electronically obtain data from each distributed sites within moments after test completion and quickly begin the data reduction and analysis process.

**Conclusions/Recommendations.** Regardless of the type of testing that is conducted, i.e., ADS versus non-ADS, the key to data reduction and analysis is developing and rehearsing a plan that sufficiently addresses the data analysis objectives of that test. Tools can be developed or acquired to meet particular data reduction and analysis needs, whether for small or large amounts of data, and whether generated by a single or multiple facilities. Rehearsal can help analysts become familiar and comfortable with the tools they will be using and ensure that the tools are capable of handling the type and amount of raw data collected so that the data can be successfully manipulated to provide meaningful SUT results. A byproduct of ADS is the communication infrastructure that allows data to be quickly moved post-test to the analysis facility.

#### **5.1.6 Measure 1-2-3-3. Degree to which ADS can increase test times, events, etc.**

**Intent.** This measure was designed to determine how ADS can increase test time and events. The time required to conduct a number of test events (trials) was compared between ADS and non-ADS phases of the EW Test.

**Data Collection and Analysis Approach.** JADS analysts reviewed detailed information contained in written test control, event, and problem logs to determine the number of trial events completed during Phase 2, as well as the amount of time spent actively testing and how that time was spent.

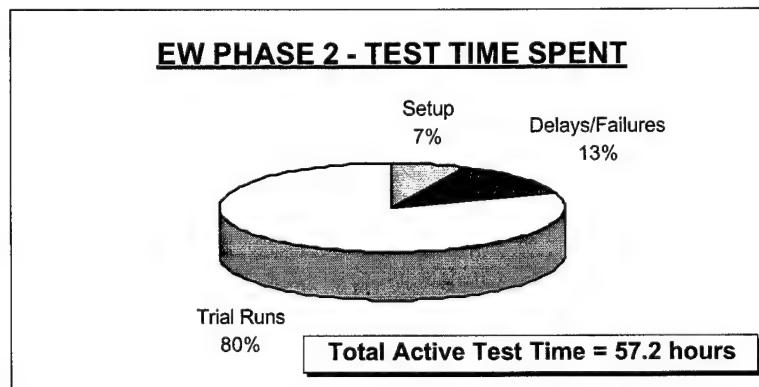
**Data Sources.** Data sources for this measure include a detailed test controller log, test event log, and a hardware, software, and network problem log (HSNPL). Information recorded in these written test logs included daily start and stop times, personnel break start and stop times, run start and stop times, run outcomes, problem start and stop times, and detailed notes on the impacts of any problems experienced.

**Results.** The total active test time during Phase 2 was 57.2 hours out of a total scheduled test time (including breaks) of 71 hours over nine days. A total of 363 trial runs were completed including 22 excursion and latency research runs and 95 aborted runs. The remaining 246 trials were considered successful test events for providing valid SUT data, resulting in an average trial success rate of 71%. Table 10 presents a summary of the trials completed during Phase 2 testing and the daily test time required. Of the time spent actively testing, .13% (7.3 hours) was lost due to delays and system failures.

**Table 10. Time Test and Run Summary**

EW PHASE 2 - TEST TIME AND RUN SUMMARY							
	TOTAL TEST TIME	ACTIVE TEST TIME	TOTAL TRIAL EVENTS	NUMBER SPECIAL TRIALS	NUMBER ABORTED TRIALS	NUMBER SUCCESSFUL TRIALS	TRIAL SUCCESS RATE
DAY 1	7:40:00	6:27:34	44	4	10	30	75%
DAY 2	7:40:00	6:07:32	44	2	10	32	76%
DAY 3	7:38:00	6:10:28	42	0	12	30	71%
DAY 4	7:45:00	6:08:00	38	3	6	29	83%
DAY 5	7:45:00	6:06:00	30	13	6	11	65%
DAY 6	7:33:30	5:33:30	36	0	12	24	67%
DAY 7	8:15:00	6:39:00	30	0	13	17	57%
DAY 8	9:00:00	7:43:00	53	0	12	41	77%
DAY 9	7:43:00	6:16:30	46	0	14	32	70%
<b>TOTALS</b>	<b>70:59:30</b>	<b>57:11:34</b>	<b>363</b>	<b>22</b>	<b>95</b>	<b>246</b>	<b>71%</b>

Figure 11 shows the percentage of active test time allocated to the following categories: test setup, delays/failures, and trial run performance.



**Note:** Test setup activities included time synchronization check and pretest communications check-out procedures.

**Figure 11. Test Setup, Delays/Failures and Trial Runs**

Table 11 compares scheduled test time and number of test events completed between Phase 2 and previous test phases including OAR and HITL.

**Table 11. Scheduled Test Time Compare to Number of Test Events Completed**

EW TEST EVENTS BY PHASE					
	SCHEDULED TEST HOURS	ACTUAL TEST HOURS	TOTAL TRIAL EVENTS	NUMBER COMPLETED TRIALS	NUMBER USABLE TRIALS
OAR	18.4 hrs	14.4 hrs*	136**	136	126
HITL	64.0 hrs	56.0 hrs	341	267	199
PHASE 2 (ADS)	71.0 hrs	57.2 hrs***	363**	246	245

\* Includes 3.5 hours of test time from the final two OAR risk reduction events

\*\* Each OAR trial provided SUT data for four active threats; data essentially equivalent to that produced by two trial events during other phases

\*\*\* Test hours and trial event totals within Phase 2 included excursion runs

**Conclusions/Recommendations.** The results discussed above show that ADS can increase the test time available and number of test events. Due to factors such as ease of scheduling and cost, almost four times as much test time was available and twice as many test events were conducted at AFEWES compared to the OAR. While it was easier to schedule test events at AFEWES, the results show that ADS did not increase the number of test events that can be conducted per unit of time.

#### **5.1.7 Measure 2-1-1-1. Degree to which live, virtual, and constructive entities exist, can be instrumented, and can be readied for a test.**

**Intent.** This measure will provide an assessment of the availability of the required ADS components to support the infrastructure (equipment, personnel, technical experts, cost, etc.) for the SPJ test.

**Data Collection Plan and Analysis Approach.** JADS performed an assessment of the RTI interface logger and the requirements needed to transfer data to the logger.

**Results.** To accurately calculate latency of all messages in the JADS EW Test federation, the development team determined that each federate needed to record data. For this reason the RTI interface logger was developed. The RTI interface logger resides between a federate and its local RTI component. It records all data that are passed to/from the RTI. Each attribute and interaction is time stamped as it goes into or comes out of the RTI. The logger was linked into each federate in the JADS EW Test federation.

The logger software was relatively easy to develop. For every function defined in the RTI interface, there was a corresponding function in the logger. It took about two man-months to develop and test the logger software. When the RTI interface specification changed, a new version of the logger had to be developed. Once the software was developed, configuring a federate to use the software was trivial. Depending on the federate design about 10 to 20 lines of

code must be modified to convert a federate to use the logger. The logger can be used with any federate as long as the RTI and logger versions match (e.g., if RTI 1.3 is used by the federate then V1.3 of the logger must also be used).

Since the file created by the logger was in binary format, the major complexity in data analysis was in writing the software that reads the log file. Although the logger can be used with any federate, special log file reader software must be written to translate the binary data to human-readable format. JADS software analysts developed log file reader software to

- Produce log file summary statistics (data counts and latency by attribute/interaction)
- Create ASCII data files used by other analysis software (e.g., ADRS)
- Create a list of threat mode changes and jammer responses
- Calculate attribute and interaction publish and receive data rates
- Display attribute and interaction header and log time values

An important part of each data type was the header. The header included a field containing the time the data were created. Latency from data creation to consumption was easily determined by calculating the difference between the header time and the log time. Another important element of the header was the sequence number. A separate sequence number was maintained for each data type within each federate. It was easy to determine data dropouts by noting missing sequence numbers for a particular attribute or interaction.

Other instrumentation needed for the AFEWES threats was provided by the facility. No unique instrumentation was needed for threats. Instrumentation to measure internal facility latency was not available for this test. This type of instrumentation was a new requirement for the facility. Instrumentation was added after Phase 2 to measure the internal latency within the AFEWES facility. These measurements will be accomplished during Phase 3.

**Conclusions/Recommendations.** Ease of instrumentation of ADS components was dependent on early planning and design for flexibility. More than a year before the simulation (federate) software was developed, the SPJ team was thinking about data collection and analysis. This planning influenced the design of the federates. Specifically, the time and sequence number were added to the data headers in the ICD to simplify the calculation of latency and determine dropouts.

Data collection requirements changed as the test progressed. The logger was developed to provide flexibility in data collection. It could be configured to log all interface events with the RTI or only certain types of events. Or it could be turned off entirely. When it was linked with a federate it required very few modifications to the federate software. It logged the data without impacting the execution of the federate. The logger could be linked with all federates in a federation and collect data at each node, be selectively linked with specific federates, or be set up as a stand-alone logger federate.

The logger was developed to collect data based on the interface to the RTI without regard for the type of data published by the federates that used the logger. This allowed the logger to be developed independently from the federate software. Changes to the logger had no impact on the

federate software. The current version of the logger took advantage of specific features of SGI computers. The logger software could easily be modified to remove the SGI features without any modification of federate software.

Instrumentation to measure internal facility latency was necessary for understanding the complete latency picture. However, traditional testing within the facility may not require this type of instrumentation so it may not be available to ADS-based tests.

#### **5.1.8 Measure 2-1-2-1. Average and peak throughput available for each link (JADS to AFEWES, JADS to ACETEF, and AFEWES to ACETEF).**

**Intent.** The intent of this measure was to provide an indication of the amount of data traffic that was sent across each link for Phase 2, as well as the amount of available bandwidth utilized to send this level of data traffic.

**Data Collection and Analysis Approach.** Data traffic over each network link was actively monitored during Phase 2 using SPECTRUM. Analysts used SPECTRUM to model the Phase 2 network and query network equipment for traffic and performance information at 30-second intervals. JADS analysts reviewed collected data for statistical trends and anomalies. Anomalies were further tracked and researched to determine any impact on collected SUT data.

**Data Sources.** The SPECTRUM tool provided a near real-time capability for network traffic monitoring, presenting current packet rate and load (percentage bandwidth utilized) information, as well as packet error and discard rate information for Phase 2 network equipment. SPECTRUM recorded captured query information to a database for later analysis.

**Results.** The average and peak packet rate and load values experienced for each Phase 2 network link are presented Table 12. These values encompass 57.2 hours of active testing over nine days.

**Table 12. Average and Peak Packet Rate and Load Values**

EW PHASE 2 - NETWORK LINK PERFORMANCE				
NETWORK LINK	PACKET RATE		BANDWIDTH UTILIZED	
	AVERAGE	PEAK	AVERAGE	PEAK
JADS - AFEWES	54.33 / sec	198 / sec	6.75%	27%
JADS - ACETEF	28.68 / sec	103 / sec	2.96%	11%
AFEWES - ACETEF	32.95 / sec	138 / sec	4.18%	19%

**Conclusions/Recommendations.** Typical data traffic, as reflected by average packet rate values, fell within expected levels for each network link. The highest levels of traffic were observed between JADS and AFEWES, corresponding to the largest amount of information that had to be shared between these two sites. The lowest traffic levels were observed between JADS and ACETEF. Average load values show that < 10 % of each T-1 line's available bandwidth was typically used to pass data between distributed sites; the maximum bandwidth utilized for any link

rose to only 27% of total capacity. While bandwidth was not an issue for JADS, future testers using ADS need to consider bandwidth early in their test design.

### **5.1.9 Measure 2-1-2-2. Percentage of complex data types received out of order by a federate.**

**Intent.** This measure was intended to determine the percentage of complex data types received in a different order than originally sent out by a federate.

**Data Collection and Analysis Approach.** The RTI interface log files were used to record all published and subscribed complex data types for each federate. JADS analysts reviewed, summarized, and compared all collected log file data. The order of all complex data types received at a federate was compared to the order in which they were published by the sending federate to determine if they were received in the same sequence or order in which they were sent.

**Data Sources.** The RTI interface loggers collected published and subscribed complex data types for each federate.

**Results.** With the available log file summary and comparison tools, no traditional examples of out-of-order data packets were discovered for Phase 2 runs. In essence, for any individual complex data type, no packets leaving a federate in one order were logged arriving in a different order at another federate. However, there were several instances of out-of-order packets according to a broader definition of out-of-order. In particular, three different types of out-of-order packets occurred.

***Out-of-Order Data Within a Federate.*** Complex data type messages generated by federate software incorporated header times to represent their creation time. Each message was also logged and time stamped by the federate logger as it left the federate. Comparison of packet header times to logger time stamps for several messages generated consecutively showed multiple examples where log file time-stamp order did not match the order in which the packets were originally generated. The most likely cause of this out-of-order data within an individual federate has been attributed to the multithreaded nature of the federate software processes themselves. Additional packet instrumentation within federates may have led to more insight into this behavior.

***Differential Delay of Complex Data Types to Receiving Federates.*** Another instance of unusual data packet ordering behavior was discovered when analysts compared the arrival times at different federates of a complex data message type generated at one federate. In many instances, the data message arrived and was logged at one site several seconds before being logged at another site. There were two general causes of this behavior. In reliable transmission, the message was passed in serial fashion from the publisher to each subscriber. In best effort, this phenomenon was due to differences in network latency between the paths. Although intuitively, this packet behavior could cause simulation visualization or other SUT performance measure anomalies, analysts did not detect any anomalous Phase 2 run behavior that corresponded to the data packet differential arrival times. Further thought on this issue brought about the

determination that no negative impact on performance measures or real-time visualization occurred because the primary visualization and data collection tool utilized during Phase 2 (ADRS) based all relevant data presentation and collection on message header time not message receipt time.

**Associated Complex Data Types Sent Via Different Transport Protocols.** Another example of oddly ordered data traffic related to the out-of-order arrival of different, yet associated, complex data types sent from a single federate to another federate. According to the Phase 2 ICD, each particular complex data message type traverses the network via a designated network transport protocols, typically TCP (reliable method) or UDP (best effort method). If a sending federate generated associated messages of different complex data types in a certain order, it is possible for them to arrive out-of-order at the receiving end because of the differential speeds of the associated transport protocol. In one particular instance, an end of data stream message was received prior to the actual last data message in the stream. Odd log file discrepancies were detected as a result of the way the RTI handled out-of-order data.

**Conclusions/Recommendations.** In the strictest sense, the Phase 2 network and RTI did not cause any out-of-order packet data. In a broader sense, however, several out-of-order packet issues were discovered through summary and comparison of the log file data collected by each federate. Some of these issues impacted packet traffic order as sent, others impacted the order received. Regardless, this analysis showed that individual federate software code, the transport protocols utilized by different complex data types, or even the particular RTI functionality selected for use in federation communication can have grave impacts on the order in which data are sent and received by federates. This out-of-order data, in turn, could cause serious SUT performance data anomalies if data collection, analysis, and real-time display tools are not developed with the potential for this in mind. Experimentation with the RTI time management functions might provide some insight into potential ways of alleviating anomalous out-of-order packet behaviors, although this RTI functionality was not implemented during Phase 2 in case unacceptably high data latencies resulted. Increased data packet instrumentation within federates (e.g., use of a data packet sniffer) would also provide more detail for further analysis into this issue.

#### **5.1.10 Measure 2-1-2-3. Percent of total complex data types subscribed to by a federate that was received by the federate.**

**Intent.** This measure was intended to report the percentage of complex data types lost while traversing the ADS network.

**Data Collection and Analysis Approach.** The RTI interface log files recorded all published and subscribed complex data types for each federate. JADS analysts utilized software tools to summarize and compare log file contents and identify lost complex data types between publishing and subscribing federates.

**Data Sources.** The RTI interface loggers collected published and subscribed complex data types for each federate. Real-time network instrumentation files and test observer notes were additional sources of information as to the potential cause of certain data losses.

**Results.** The analysis of lost data was approached from both a number of lost messages standpoint and a time standpoint. For the first approach, lost messages were tallied for six complex data message types across relevant network links. The six types were selected for evaluation based on their ability to provide insight into the impact of lost traffic on SUT data validity. In other words, these were the message types that, if lost, should have had the most noticeable effect on SUT behavior and the SUT performance measure data collected. For these message types, there were no TCP (reliable) data traffic losses, and <1% of all UDP (best effort) data traffic was lost. Thirty-eight individual runs with unusual data losses (> 5 messages lost) were marked and further studied for anomalies before being included in the SUT valid data set. Table 13 provides a summary of lost data traffic categorized by message type and network link.

**Table 13. Lost Data Traffic Messages by Link**

DATA ELEMENT	TYPE	JADS-AFEWES	JADS-ACETEF	AFEWES-ACETEF
Live Entity State (A/C TSPI)	UDP	Avg Lost: 6.9 Avg Sent: 4000 Percent Lost: < 1 Max: 503	Avg Lost: 18.6 Avg Sent: 4000 Percent Lost: < 1 Max: 1266	N/A
Threat Performance (Threat Track Data)	UDP	Avg Lost: 6.3 Avg Sent: 4000 Percent Lost: < 1 Max: 504	Avg Lost: 18.1 Avg Sent: 4000 Percent Lost: < 1 Max: 1266	N/A
Threat Performance (T/E, J/S, Target Loc)	UDP	Avg Lost: 4.2 Avg Sent: 4000 Percent Lost: < 1 Max: 182	N/A	Avg Lost: 14.8 Avg Sent: 4000 Percent Lost: < 1 Max: 2222
SUT_Jammer_Tech (DSM RF Emissions)	TCP	N/A	Avg Lost: 0 Avg Sent: 9 Percent Lost: 0 Max: 0	Avg Lost: 0 Avg Sent: 9 Percent Lost: 0 Max: 0
SUT_Receiver_Track (Verify Environment)	TCP	N/A	Avg Lost: 0 Avg Sent: 96 Percent Lost: 0 Max: 0	N/A
Source_Mode Change (Threat RF Emission)	TCP	Avg Lost: 0 Avg Sent: 50 - 90 Percent Lost: 0 Max: 0	N/A	Avg Lost: 0 Avg Sent: 50 - 90 Percent Lost: 0 Max: 0

A/C = aircraft

T/E = tracking error

For the second approach, analysts detected all data losses longer than one second and attempted to categorize the cause and outcome of each data loss using a combination of observer notes and test instrumentation. The outcome of individual data losses was typically one of two extremes; either the loss had essentially no observable impact on SUT performance measure data or the run

had to be aborted. This data loss outcome was dependent on several factors including the duration of the outage and the associated number, type, destination, and importance of the lost message packets. The timing of the outage during the run was noted to have an impact as well. For instance, an 8 to 10 second TSPI data loss during a run could be overcome by federation software dead reckoning algorithms, while a similar loss at the beginning of a run, before the transfer of any TSPI data, could not be handled and resulted in a manual abort of the run. The cause of each data loss event turned out to be a tougher analysis problem. Instrumentation sources included network equipment self-diagnostic error message files, observed real-time ping traffic outages, and real-time network load and packet rate query results; yet, even with this instrumentation many of the data losses that occurred during Phase 2 remained impossible to attribute definitively to network links, network equipment, the RTI, or federation activity. In most cases, especially those where the data loss duration was short, there was just not conclusive evidence to determine the outage cause.

**Conclusions/Recommendations.** No TCP (reliable) data traffic losses were detected. However, the nonunique sequence numbering problem experienced in Phase 2 limited the ability of analysts to verify that no reliable data traffic was lost. Less than 1% of all UDP (best effort) data traffic was lost. Of the thirty-eight individual runs with unusual UDP data losses ( $> 5$  messages lost) that were marked for further study, only one run was actually excluded from the SUT valid data set, since a several second data loss resulted in anomalous tracking error and missile miss distance data points. The SUT performance measure data from the other thirty-seven runs were shown to not differ significantly from the remaining run data collected. Detailed analysis of each data loss event did not always result in successful determination of the cause of the event, although some data losses were obviously attributable to the occasional network link and network equipment problems identified during Phase 2. Increased data packet instrumentation within federates, perhaps via the use of a data packet sniffer, would provide more detail for further analysis into the cause of data loss events. JADS was able to overcome most of the data losses through simple dead reckoning at AFEWES and in the DSM coupled with the combination of aircraft velocity (360 knots), and flight profile (straight and level). High-speed aircraft and high maneuver rates would increase the complexity of the dead reckoning that needs to be used to minimize the correction once data flow resumed. This was especially true for bursts of data loss as large as those measured during this test. Reducing the size of the data loss bursts may also reduce the complexity of the dead reckoning algorithm. Designers need to be aware that data loss will occur and plan accordingly.

#### **5.1.11 Measure 2-1-2-4. Average and peak data latency.**

**Intent.** This measure was intended to report the average and peak latency experienced by test data elements traversing the ADS network.

**Data Collection and Analysis Approach.** The RTI interface log files recorded the arrival and departure times of all published and subscribed complex data types for each federate. JADS analysts utilized software tools to summarize and compare log file contents and determine round-trip federation latency as well as node-to-node latency values for particular complex data types.

**Data Sources.** The RTI interface loggers at each federate recorded the arrival and departure times of all published and subscribed complex data types. A data packet sniffer installed to monitor the ACETEF federated traffic during six runs on 4 Dec 98 provided some additional insight into the latency issue.

**Results.** The analysis of high latency was also approached in two ways, one focusing on node-to-node latency while the other focused on latency for just those federation messages deemed latency critical. According to the Phase 2 ICD, latency critical message types include the *MS\_Source\_Mode\_Change*, *SUT\_Jammer\_Tech\_Com*, and *SUT\_Receiver\_Track\_Update* complex data message types. Latency values were computed for these message types to travel round trip between the AFEWES and ACETEF federates. For the first approach, node-to-node latency values across relevant network links were evaluated for six complex data message types. The six types were selected for evaluation based on their ability to provide insight into the impact of latent traffic on SUT data validity. In other words, these were the message types that, if latent, should have had the most noticeable effect on SUT behavior and the SUT performance measure data collected. Ten individual runs with unusually high node-to-node latency values, out of 246 completed runs, were marked and further studied for anomalies before being included in the SUT valid data set. Table 14 provides a summary of node-to-node latency categorized by message type and network link.

**Table 14. Node-to-Node Traffic Latency by Data Element (milliseconds)**

DATA ELEMENT	TYPE	JADS-AFEWES	JADS-ACETEF	AFEWES-ACETEF
Live Entity State (A/C TSPI)	UDP	Avg: 43.9 Max: 859	Avg: 41.2 Max: 861	N/A
Threat Performance (Threat Track Data)	UDP	Avg: 45.9 Max: 860	Avg: 42.6 Max: 861	N/A
Threat Performance (T/E, J/S, Target Loc)	UDP	Avg: 32.1 Max: 7975	N/A	Avg: 35.7 Max: 8540
SUT_Jammer_Tech (DSM RF Emissions)	TCP	N/A	Avg: 130 Max: 13951	Avg: 104.3 Max: 9680
SUT_Receiver_Track (Verify Environment)	TCP	N/A	Avg: 112.7 Max: 13982	N/A
Source_Mode Change (Threat RF Emission)	TCP	Avg: 101 Max: 9556	Avg: 66 Max: 8022	Avg: 61 Max: 7701

A/C = aircraft

T/E = tracking error

For the second approach, analysts calculated round-trip federation latency values by summing the individual message latencies between AFEWES and ACETEF nodes for latency critical message types. Out of 246 successfully completed trial runs, eight experienced unsuitable round-trip federation latency values ( $> 500$  ms) and were marked for probable exclusion from the valid SUT data set. Calculated average and maximum round-trip federation latencies based on the remaining successful runs, were 254 milliseconds (ms) and 380 ms, respectively. Further analysis was performed on the marked runs to determine the potential causes and outcomes of extremely high (e.g.,  $>13$  seconds) latency values. Two causes were determined to be responsible for these extremes. The first, a network link or network equipment outage, which caused reliable (TCP)

data traffic to be held up on multiple occasions, was deemed responsible for < 15% of the extreme latency values seen. The other, more serious problem was due to a design defect in the DSM federation software code that has since been identified and fixed. During Phase 2, an algorithm (i.e., Nagle algorithm) was in the code that caused some message traffic to wait and be bundled with later traffic before being distributed by the federate. Depending on the order and timing in which latency critical message types were bundled for distribution, they may have had to experience large wait times.

**Conclusions/Recommendations.** High round-trip federation and node-to-node latencies impacted 18 out of 246 successfully completed test trials. Thus, approximately 7% of the trials had to be more carefully researched to determine if high latency resulted in SUT data anomalies. Of these eighteen individual runs, only one run was actually excluded from the SUT valid data set, since a approximately 14 second latency for reliable data traffic, in conjunction with several seconds of data loss, resulted in anomalous tracking error and missile miss distance data points. The SUT performance measure data from the other seventeen runs were shown to not differ significantly from the remaining run data collected. Detailed analysis of individual latency events resulted in an understanding of the problems with the Nagle algorithm being implemented in federation code, as well as the potential for short duration network outages to increase reliable traffic message latency. Installation of the data packet sniffer at just one site on one day enabled more detailed and insightful analysis for on few runs.

JADS avoided latency impacts in most of the EW Test MOPs by making all measurements for that MOP inside the AFEWES facility. Only response time and time to correct ID were measured across two facilities. Measuring within a single facility or frame of reference was the preferred approach to removing the impact of latency from the MOP. Even so, high-speed aircraft and high rates of maneuver increased differences in each facility's point of view. Higher latency between the facilities made the problem worse. This was not a problem for open-loop tests for obvious reasons. Closed-loop tests can become unrealistic if the differences are too great. Test designers need to understand this and design tests where the closed-loop interaction is in one frame of reference only, or they need to set hard limits on the allowable latency based on the test objectives and players.

#### **5.1.12 Measure 2-1-3-1. Degree to which test events (trials) were affected by ADS components (failure or otherwise) exclusive of network problems.**

**Intent.** The intent of this measure was to determine the impact of ADS component availability on test trial events.

**Data Collection and Analysis Approach.** A HSNPL was used in conjunction with test control logs, event logs, and site notes to document ADS component problems and aborted runs. JADS analysts reviewed, categorized, and summarized the number, type, and duration of problems encountered to determine the number of test events (trials) impacted by ADS components.

**Data Sources.** A HSNPL was used to document ADS component problems, as were test control logs, event logs, and site observer notes. Information recorded in these written logs included

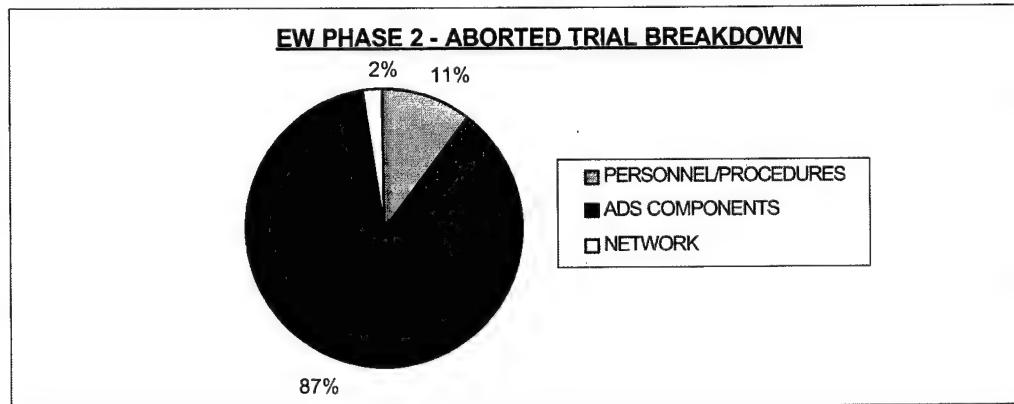
notes on all problems experienced, problem start and stop time, and the particular trial events impacted.

**Results.** ADS component problems were encountered frequently during Phase 2 and were responsible for almost all of the test time lost due to delays and failures (7.3 hours lost out of 57.2 hours total active test time). Specific problems included numerous federate crashes, real-time analysis tool (ADRS) crashes, AFEWES equipment software and mechanical problems, data dropouts, federation setup and script problems, and secure voice communications equipment malfunctions. Table 15 summarizes the impact of ADS component problems on trial events.

**Table 15. Impact of ADS Component Problems**

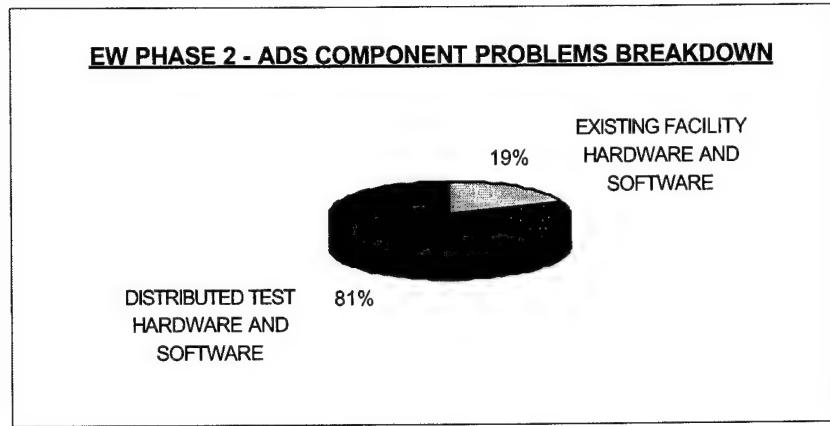
TRIALS LOST DUE TO ADS COMPONENT PROBLEMS	
TOTAL TEST TRIALS	363
TOTAL ABORTED TRIALS	95
ADS COMPONENT PROBLEMS FAULTED IN ABORTED TRIAL:	83
- FEDERATE AND ANALYSIS TOOL CRASHES	48
- FEDERATION SETUP AND SCRIPT PROBLEMS	5
- COMMUNICATION TOOL AND DATA DROPOUTS	14
- AFEWES MECHANICAL AND SOFTWARE PROBLEMS	16

Figure 12 shows the breakdown of aborted trial runs by fault category. ADS component problems were responsible for 83 lost runs (87%) during Phase 2.



**Figure 12. Aborted Trial Breakdown**

Figure 13 shows lost runs due to ADS component problems further separated into two categories in order to provide additional insight into the impacts of existing test equipment failures versus failures attributable to system hardware and software implemented specifically to enable distributed testing (e.g., RTI software.)



**Figure 13. ADS Component Problems Breakdown**

Although the majority (81%) of the runs aborted due to ADS component problems were aborted as a result of distributed test hardware and software malfunction, the actual amount of test time lost to those problems was only 1.75 hours, as compared to 4.3 hours lost due to problems with existing facility hardware and software components.

**Conclusions/Recommendations.** ADS component problems, such as numerous federate crashes, real-time analysis tool (ADRS) crashes, AFEWES equipment problems, data dropouts, federation setup and script problems, and voice communications equipment malfunctions were responsible for the majority of lost test time and resulted in 83 aborted runs. Approximately 19% of these aborted runs (16 runs; 4.3 test hours) were lost because of problems with existing test system equipment (primarily at AFEWES), while the remaining 81% (67 runs; 1.75 test hours) were lost because of problems with equipment implemented to enable distributed testing. Designers need to understand that additional components add both complexity and down time. Component reliability is key to efficiency.

#### **5.1.13 Measure 2-1-3-2. Degree to which test events (runs) were affected by network problems (failure or otherwise).**

**Intent.** The intent of this measure was to determine the impact of ADS network availability on test trial events. The network system included all software and hardware used for connecting the distributed sites between routers.

**Data Collection and Analysis Approach.** A HSNPL was used in conjunction with test control logs, event logs, and site notes to document ADS network problems and aborted runs. JADS analysts reviewed, categorized, and summarized the number, type, and duration of problems encountered to determine the number of test events (trials) impacted by the ADS network.

**Data Sources.** A HSNPL was used to document ADS network problems, as were test control logs, event logs, and site observer notes. Information recorded in these written logs included

notes on all problems experienced, problem start and stop times, and the particular trial events impacted.

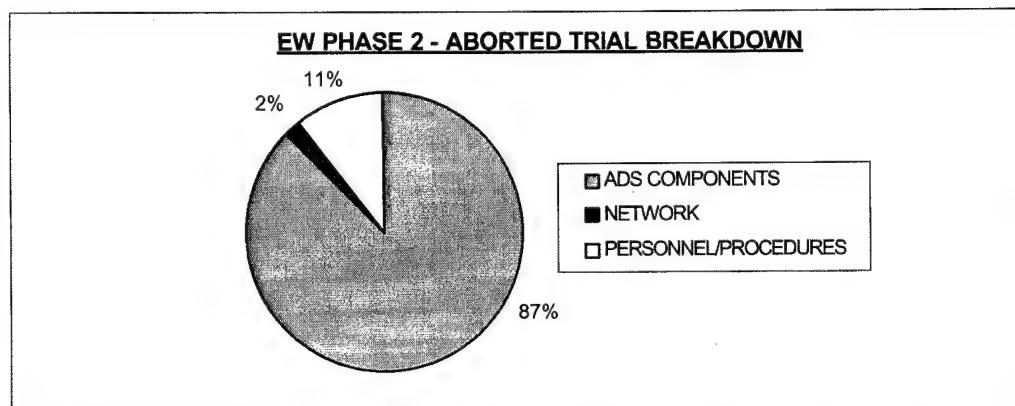
**Results.** Network problems encountered during Phase 2 were limited and resulted in just a few minutes of actual lost test time. Specific problems included a bad IDNX voice card at ACETEF, other ACETEF router problems, and a momentarily down link between ACETEF and AFEWES. Table 16 summarizes the impact of ADS network problems on trial events.

**Table 16. Impact of ADS Network Problems on Trial Events**

TRIALS LOST DUE TO ADS NETWORK PROBLEMS	
TOTAL TEST TRIALS	363
TOTAL ABORTED TRIALS	95
ADS NETWORK PROBLEMS FAULTED	2 *

\* Network problems resulted in two additional trials being run using nonsecure voice communication.

Figure 14 shows the breakdown of aborted trial runs by fault category. ADS network problems were responsible for 2% of the lost runs during Phase 2.



**Figure 14. Aborted Trial Breakdown**

**Conclusions/Recommendations.** ADS network problems, including several ACETEF router and IDNX voice card problems, were responsible for just a few minutes of lost test time and resulted in only two aborted runs. This was not surprising considering the reliability of most computer networks and wide area network hardware.

#### **5.1.14 Measure 2-1-3-3. Degree to which test events (trials) were affected by personnel problems.**

**Intent.** This measure was intended to identify the impacts of personnel problems including problems related to training, manning, consistency, and coordination on test trial events. This

measure was intended to collect data on the human element of an ADS test and the impacts associated with human error and human creativity.

**Data Collection and Analysis Approach.** A HSNPL was used in conjunction with test control logs, event logs, and site observer notes to document personnel and procedural problems and aborted runs. JADS analysts reviewed, categorized, and summarized the number, type, and duration of problems encountered to determine the number of test events (trials) impacted by personnel.

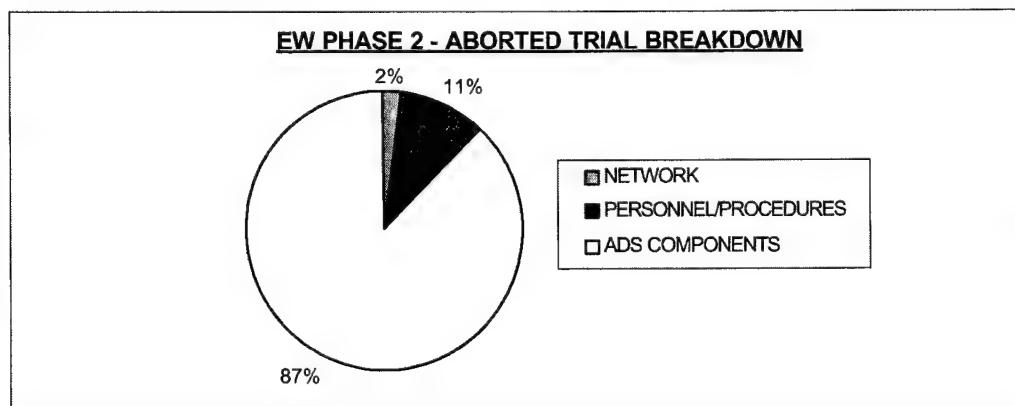
**Data Sources.** A HSNPL was used to document problems encountered with personnel and procedures, as were test control logs, event logs, and site observer notes. Information recorded in these written logs included notes on all problems experienced, problem start and stop times, and the particular trial events impacted.

**Results.** Personnel and procedural problems encountered during Phase 2 included script problems, miscommunication among test controllers and test station operators, operator tardiness, and operator slowness or unfamiliarity with changed procedures. Table 17 summarizes the impact of ADS component problems on trial events.

**Table 17. Impact of ADS Component Problems on Trial Events**

TRIALS LOST DUE TO PERSONNEL AND PROCEDURAL PROBLEMS	
TOTAL TEST TRIALS	363
TOTAL ABORTED TRIALS	95
PERSONNEL/PROCEDURES FAULTED	10

Figure 15 shows the breakdown of aborted trial runs by fault category. Personnel and procedural problems were responsible for 11% of the lost runs during Phase 2.



**Figure 15. Aborted Trial Breakdown by Fault Category**

**Conclusions/Recommendations.** Difficulties with personnel and procedures including loading correct scripts, ensuring scripts were located in correct directories, miscommunication among test controllers and test station operators, operator tardiness, and operator slowness or unfamiliarity with changed procedures were responsible for the second largest portion of lost test runs and resulted in 10 aborted runs.

**5.1.15 Measure 2-2-1-4. Ease with which data can be retrieved, post-trial, from a given node.**

**Intent.** The intent of this measure was to determine the degree of difficulty in retrieving ADS data from the distributed EW Test nodes.

**Data Collection and Analysis Approach.** Interviews were conducted with Phase 2 data managers and analysts to ascertain their perceptions about ADS post-trial data retrieval. Interview questions focused on identifying particular problems noted using retrieval procedures or tools. JADS analysts reviewed and summarized recorded remarks.

**Data Sources.** Seven interviews were conducted. Interviewees included the TCAC data manager, the ACETEF data management representative, and five JADS data analysts.

**Results.** Interviewees listed data retrieval methods and tools that were used during Phase 2 for bringing distributed site data to the TCAC for analysis. Methods ranged from handcarrying written observer logs to the electronic transfer of large data files using UNIX file transfer protocols. Most discussion focused on electronic transfer methods. The federate log files and other data of interest from all distributed sites were transferred daily to the TCAC (after test completion) in less than an hour. Most respondents admitted that the data retrieval procedures were not formally documented and that improvement could be made in that area. Some frustration was caused for test participants who were unaware that their written observer logs would be retrieved after testing, and for others who found no plan for retrieving data (e.g., distributed site database files, configuration management and software documentation) they felt was necessary to the analysis effort. One other issue stemmed from the fact that electronic transfer of data files over the network increased utilized network bandwidth to near 100%, a level that would indicate a problem during typical testing. Since an occasional unscheduled data retrieval took place between test trials, monitored network load data had to be manually filtered to eliminate any atypical increases before analysis could be done in that area.

**Conclusions/Recommendations.** In general, respondents indicated that, while not well documented, the data retrieval methods implemented during Phase 2 met objectives. Recommendations included better planning and documentation of data retrieval processes, as well as implementation of a more well-defined directory structure for organizing all the retrieved data.

It should be noted that it took up to ten days after traditional test events in Phase 1 to obtain the electronic data from the test facility.

### **5.1.16 Measure 2-2-2-1. Degree to which test managers can control the configurations of ADS participants, the ADS environment data, and ADS networks.**

**Intent.** This measure was intended to assess the ability of the test manager to adequately control the configuration of ADS participants, the ADS environment data, and ADS networks both during and between test events.

**Data Collection and Analysis Approach.** Interviews were conducted with the EW Test manager and test controllers to ascertain their perceptions about ADS configuration control. Interview questions required assessment of documents, tools, and reports used to establish configuration control as well as the overall effectiveness of configuration control procedures. Interviewees were asked to make recommendations for future improvement of the configuration control process. JADS analysts reviewed and summarized recorded remarks.

**Data Sources.** Five interviews were conducted. Interviewees included the EW Test manager, the two JADS test controllers, the ACETEF test executor, and the network analysis station operator.

**Results.** Although a few existing configuration reports were mentioned (e.g., usable script lists, acceptance test software library lists), the general consensus was that configuration control of ADS component software, networks, and environment data prior to and during testing was limited. The EW Test manager was not in direct control over the software and system version implemented but was forced to rely on the developers configuration management methods. Development in one facility and integration in another seemed to especially stress configuration management. Problems caused by constantly changing test component software and test tools were encountered throughout integration, acceptance testing, and actual testing. Dynamic processes controlled by RTI federation execution software initiated environment data problems during these test events, which were also out of the realm of JADS control. Although configurations were finally stabilized enough to permit successful Phase 2 test execution, most respondents did not feel comfortable with the level of configuration control.

**Conclusions/Recommendations.** In general, respondents indicated that, while established configuration control procedures did enable Phase 2 to be carried out successfully, configuration control procedures for ADS participants and environment data were not satisfactory. Recommendations included allocating more control to the test manager and formalizing documentation and procedures utilized by contracted support personnel. Configuration management is essential to ADS testing. Different configuration management practices and standards across facilities are often a reality. The potential for uncontrolled change increases as the number of sites increases. More potential exists when the facility supports more than one customer during the integration and actual test period. HLA adds additional configuration items. The test organization needs to be able to handle this increased complexity.

### **5.1.17 Measure 2-3-2-3. Degree to which protocols, processes, and procedures are needed to enable effective centralized test control.**

**Intent.** The measure was intended to determine the degree to which protocols, processes, and procedures were needed to enable effective centralized test control.

**Data Collection and Analysis Approach.** Interviews were conducted with Phase 2 test controllers and test executors to ascertain their perceptions about effective centralized test control for ADS. Interview questions focused on the effectiveness of documented test control procedures, as well as the need for modifying and adding procedures during a test. Respondents were asked to identify particular areas of difficulty and potential fixes. JADS analysts reviewed and summarized recorded remarks.

**Data Sources.** Eight interviews were conducted. Interviewees included the two JADS test controllers in the TCAC, the two JADS test executors at AFEWES and ACETEF, and four test station operators in the TCAC.

**Results.** The interview responses given by personnel in the TCAC (including test controllers and station operators) indicated strong feelings of effective centralized test control. Test executors at the distributed sites, on the other hand, did not rate the effectiveness of centralized test control quite as high. On a 1-6 scale, representing the spectrum from ineffective centralized control to effective centralized control, all TCAC respondents provided ratings of 5 and 6, while the two distributed respondents provided ratings of 3 and 4. Modifications to checklist procedures and communications processes, while enhancing the overall efficiency of the testing, made distributed test executors feel temporarily “out-of-the-loop” as they struggled to adjust, particularly during personnel changes. All interviewees but one agreed that documented procedures aided in the establishment of effective, centralized test control, and that modifications to federate initialization and communications processes alleviated many problems encountered during rehearsals and initial testing. Miscommunication within and among sites due to poorly defined processes or malfunctioning communications hardware was identified as the biggest problem area for effective control.

**Conclusions/Recommendations.** In general, respondents felt that documented or modified protocols, processes, and procedures enabled effective centralized test control to take place. Recommendations included improved voice communications equipment and stricter, more well-defined communications processes.

#### **5.1.18 Measure 2-3-2-4. Degree to which real-time analysis systems support test safety and other test control requirements.**

**Intent.** This measure was intended to determine what real-time analysis was required and the impact of having real-time analysis systems for test control.

**Data Collection and Analysis Approach.** Interviews were conducted with Phase 2 test controllers and test executors to ascertain their perceptions about having real-time analysis systems for test control. Interview questions focused on the manner in which real-time analysis tools provided feedback that improved the test controller’s ability to control test events. JADS analysts reviewed and summarized recorded remarks.

**Data Sources.** Seven interviews were conducted. Interviewees included the two JADS test controllers in the TCAC, the two JADS test executors at AFEWES and ACETEF, and three test station operators in the TCAC.

**Results.** Interview responses were unanimous in agreement that the feedback provided by real-time analysis tools improved the ability of test controllers to control test events. Such tools allowed test controllers to watch distributed events unfold according to expectation from a central facility and gave nearly immediate identification of distributed component problems and data losses. They provided a cursory feel for the usefulness of each test trial and enabled test controllers to make timely decisions about trial events or system problems.

**Conclusions/Recommendations.** Real-time analysis tools greatly enhanced the ability of ADS test controllers to control test events and provided greater situational awareness to other test participants. No conclusion was made about the ability of such tools to impact test safety, as there were no safety issues identified for Phase 2.

## **6.0 Correlation Analysis**

Although the emphasis of Phase 2 was on the impact of ADS on the test components rather than on the performance of the test item itself, EW Test MOPS were computed as a means of determining potential ADS impacts. JADS hoped differences between data sets collected during ADS testing and baseline data collected by more traditional test means in the OAR and HITL phases would point analysts to areas where ADS testing had significant impacts; likewise, similarities between ADS and non-ADS phase data sets were to confirm minimal ADS impact. It was expected that Phase 2 MOP data would not differ significantly from baseline data for most threat systems and reference test conditions. However, the unexpectedly large influences on the data collected from all EW Test phases because of operator variance and differing threat system representations among facilities hampered the ability of analysts to clearly realize the impacts of ADS, sometimes skewing data sets in extraordinary ways.

### **6.1 EW Test Measure of Performance (MOP) Evaluation**

During Phase 2, SUT performance data were collected for the ten test measures listed in Table 1. The evaluation process for each MOP included sorting the collected raw data into sets by threat system and reference test condition, determining the distribution shape and parameters for each data set, calculating descriptive statistics, and correlating each Phase 2 data set to the OAR and HITL baseline data. All Phase 2 classified descriptive statistics and frequency histograms showing the shape, central tendency, and dispersion, or variance of the MOP data sets are published under separate cover. The following sections detail the correlation process and the correlation analysis results performed between matching data sets (i.e., data collected during different test phases under the same reference test condition) using statistical hypothesis testing.

### **6.2 Statistical Hypothesis Testing**

Two sample data sets are said to correlate, or equate, if it can be determined that they are analogous with respect to certain distribution parameters. Statistical hypothesis testing provides a means of determining how well the distribution shape, location, and dispersion parameters of two data sets equate. For each distribution parameter, an appropriate statistical comparison test was selected based on the distribution form of the collected data (e.g., binomial, normal). The underlying hypothesis of each test is that the two data samples are equivalent; that is, they represent the same true population. If, in performing the test, this hypothesis could not be rejected with reasonable confidence, then the two data sets were determined to correlate.

Numerous statistical hypothesis tests are available for comparing the distribution parameters of sample data sets. However, valid application of any one of these tests necessitates meeting its underlying assumptions and data requirements. For example, some tests are invalid when applied to non-normal or non-continuous data; others may require a minimum number of samples to provide power in distinguishing between data sets. Successful application lies in choosing a test that is both valid and powerful in determining the extent to which the data sets correlate. Four

comparison tests with wide utility were chosen for application in MOP correlation analysis after the characteristics of the data collected during initial EW Test phases were assessed. These included a comparison of proportions of the test applicable to binomially distributed (pass/fail) data; a T-test or means comparison test for comparing location parameters for roughly normal shaped data; an F-test or variance test for comparing the dispersion of roughly normal shaped data; and a Kolmogorov-Smirnov (K-S) test for correlating the overall shape, including the mean and variance, of data sets that may not meet an underlying assumption of normality.

Regardless of the type of hypothesis test selected, the methodology for performing the test was standard. For each, a test statistic value was computed by inserting the collected data values from the two samples into a mathematical equation. However, it is not possible, using statistical hypothesis testing, to conclude that two sets of sample data were the same with absolute certainty. Instead, the underlying hypothesis must be accepted or rejected based on the probability of obtaining the generated test statistic, along with the tester's willingness to risk making an incorrect conclusion. If the underlying hypothesis is true and the two data sets are indeed from the same population, it is highly probable that the test statistic value generated will fall within an expected range. If it falls outside this range, (i.e., is more extreme), then it is more likely that the two samples are not analogous for that distribution parameter. For each type of hypothesis test, there is a statistical table that associates a probability value, or P-value, with the generated test statistic value based on the range of values expected when the underlying hypothesis is true. This P-value is the result value reported, (e.g., P-value = .0452). In statistical terms, the P-value represents "the probability of attaining the given value of the test statistic or a more extreme value if the null hypothesis is true." To use the P-value, it must be compared to the level of risk that the tester is willing to take in incorrectly rejecting the underlying assumption. Essentially, the underlying hypothesis should only be rejected if the tester is comfortable with a level of risk greater than the P-value reported. If the tester is only willing to risk a 5% chance of an incorrect rejection, then given any P-value  $< .05$ , the underlying hypothesis should be rejected. If a 10% risk is acceptable, the underlying hypothesis may be rejected for P-values up to .10.

### 6.3 Correlation Results

This section contains the results of the correlation tests performed on each of the MOP data sets. Each available data set within a reference test condition (north dry, north wet, south dry and south wet) was correlated against every other data set. The resultant P-values from each of the three correlation tests (T-test, F-test, and K-S test) are shown in Tables 18-25.

To reference each test condition, use the following abbreviations.

**P2** - data set collected using Phase 2 data; latency is included where appropriate in all measurements.

**P2L** - response time data collected during Phase 2 with latency removed.

**HITL** - data set collected using HITL data prepared by GTRI; latency is included where appropriate in all measurements.

**OAR** - data set collected using OAR data prepared by GTRI; latency is included where appropriate in all measurements.

When referencing the columns in the following tables, the two labels shown are the two data sets compared in that correlation test.

To view the actual data points used in these correlation tests, please reference the classified EW Test results report. This report contains details about the methods used in collecting each data set for each MOP and also shows the relative and cumulative frequency histograms for each data set used in the correlation tests.

**Table 18. Correct ID Response Time Correlation Matrix**

	System 1			System 2			System 3			System 4		
Test	P2L-P2	P2L-SIL	P2-SIL									
<b>North Wet</b>												
T test	*	*	.0000	.0000	.0590	.0000	.0000	.2044	.9879	.1141	.1334	.0134
F test	*	*	.0000	.0000	.0000	.0000	.1022	.0000	.0000	.5041	.0000	.0000
KS Test	.0000	.0000	.0000	.0000	.0000	.0000	.0013	.0001	.0004	.0167	.0020	.0001
<b>South Wet</b>												
T test	.0000	.4683	.0000	.0002	.2999	.0049	*	*	.6962			
F test	.0000	.0000	.0000	.2918	.0000	.0000	*	*	.0000			
KS Test	.0000	.0001	.0000	.0000	.0007	.0003	.0000	.0000	.0000			

NOTE: \*One of the two compared data sets had no variance, so P-value could not be computed.

**Table 19. Correct ECM Technique Selection Response Time Correlation Matrix**

Test	System 1			System 1			System 1		
	P2L-P2	P2L-SIL	P2-SIL	P2L-P2	P2L-SIL	P2-SIL	P2L-P2	P2L-SIL	P2-SIL
<b>North Wet</b>									
<b>T test</b>	*	*	.0000	.0000	.0550	.0000	.0000	.6427	.4194
<b>F test</b>	*	*	.0000	.0000	.0000	.0000	.1030	.0000	.0000
<b>KS Test</b>	.0000	.0000	.0000	.0000	.0000	.0000	.0003	.0008	.0004
<b>South Wet</b>									
<b>T test</b>	.0170	.0163	.0000	.0000	.0000	.0957	*	*	.3832
<b>F test</b>	.4312	.0004	.0002	.0636	.4085	.1037	*	*	.0000
<b>KS Test</b>	.0000	.0001	.0000	.0000	.0000	.0004	.0000	.0000	.0000

NOTE: \*One of the two compared data sets had no variance, so P-value could not be computed.

**Table 20. RMS Tracking Error Correlation Matrix**

	System 1			System 2			System 3			System 4		
Test	P2-HITL	P2-OAR	HITL-OAR									
<b>North Dry</b>												
T test	.1474	.2179	.1024	.8025	.5594	.6691	.0000	.1222	.0000	.0844	.0030	.0000
F test	.0000	.0000	.0000	.4444	.0030	.0054	.0000	.3479	.0001	.0000	.2400	.0000
KS Test	*	*	*	*	*	*	*	*	*	*	.0000	*
<b>South Dry</b>												
T test	.5092	.3137	.2804	.5960	.9116	.3965	.7421	.0335	.0515			
F test	.4124	.0000	.0000	.0015	.1029	.0288	.0559	.3087	.0159			
KS Test	*	*	*	*	1.000	*	*	*	.0088			
<b>North Wet</b>												
T test	.0017	.0022	.0215	.0465	.0404	.6117	.0014	.2566	.0373	.0008	.0176	.0060
F test	.0000	.0000	.0000	.4333	.0005	.0007	.0000	.0000	.0000	.0000	.0000	.0000
KS Test	.0000	.0000	.0000	.1880	.5978	.8981	.0004	.0045	1.000	.0001	.0044	.0000
<b>South Wet</b>												
T test	.0000	.0000	.1572	.9807	.2052	.1886	.0743	.2209	.0116			
F test	.0000	.0000	.3607	.2968	.1544	.0547	.0000	.0007	.0000			
KS Test	.0000	.0000	.0173	.9386	.0716	.0030	.0170	.0003	.0000			

NOTE: \*One of the two compared data sets had no variance, so P-value could not be computed.

**Table 21. Jamming-to-Signal Ratio Correlation Matrix**

	System 1	System 2	System 3	System 4
Test	P2-HITL	P2-HITL	P2-HITL	P2-HITL
<b>North Wet</b>				
T test	.0000	.0000	.0000	.0000
F test	.0000	.0000	.0000	.0000
KS Test	.0000	.0000	.0000	.0000
<b>South Wet</b>				
T test	.0000	.0000	.0000	
F test	.0000	.0000	.0000	
KS Test	.0000	.0000	.0000	

**Table 22. Number of Breaklocks Correlation Matrix**

Test	System 1			System 3			System 4		
	P2-HITL	P2-OAR	HITL-OAR	P2-HITL	P2-OAR	HITL-OAR	P2-HITL	P2-OAR	HITL-OAR
<b>North Dry</b>									
T test	*	.2874	*	.1792	.9095	.0953	.8787	.0001	.0001
F test	*	.2256	*	.0078	.2215	.0396	.4562	.0003	.0026
KS Test	**	**	**	**	**	**	**	**	**
<b>South Dry</b>									
T test	*	.0409	*	.9206	.0085	.2614			
F test	*	.1193	*	.3947	.0312	.0659			
KS Test	**	*	**	**	**	**			
<b>North Wet</b>									
T test	.0258	.0000	.0000	.0038	.3635	.0004	.0048	.0036	.2647
F test	.1530	.0000	.0000	.0005	.1939	.0000	.0868	.0002	.0123
KS Test	.0663	.0000	.0000	.1185	1.000	.0168	.0050	.0406	.5884
<b>South Wet</b>									
T test	.0005	.0000	.0000	.0016	.0834	.0000			
F test	.0000	.0000	.0000	.0000	.0125	.0000			
KS Test	.0177	.0000	.0000	.0588	.8446	.0006			

NOTE: \*One of the two compared data sets had no variance, so P-value could not be computed.

\*\*One or both of the compared data sets had less than 16 samples, so P-value is not computed

**Table 23. Reduction in Engagement Time Correlation Matrix**

Test	System 1			System 3			System 4		
	P2-HITL	P2-OAR	HITL-OAR	P2-HITL	P2-OAR	HITL-OAR	P2-HITL	P2-OAR	HITL-OAR
<b>North Wet</b>									
<b>T test</b>	.0000	.0000	.0000	.0000	.2494	.0168	.0810	.4114	.9420
<b>F test</b>	.1161	.0000	.0000	.0000	.0000	.0000	.1720	.0000	.0000
<b>KS Test</b>	.0000	.0000	.0000	.0000	.0000	.6236	.0583	.0095	.0030
<b>South Wet</b>									
<b>T test</b>	.0000	.0000	.0000	.0746	.0002	.9853			
<b>F test</b>	.0054	.0000	.0000	.0000	.2856	.0000			
<b>KS Test</b>	.0004	.0000	.0022	.8495	.1070	.3894			

NOTE: \*One of the two compared data sets had no variance, so P-value could not be computed.

**Table 24. Reduction in Missiles Launched Correlation Matrix**

Test	System 1			System 3		
	P2-HITL	P2-OAR	HITL-OAR	P2-HITL	P2-OAR	HITL-OAR
<b>North Wet</b>						
<b>T test</b>	.0000	.0000	.0000	*	*	.0464
<b>F test</b>	.0108	.0000	.0000	*	*	.0000
<b>KS Test</b>	.0000	.0000	.0001	1.000	1.000	.4677
<b>South Wet</b>						
<b>T test</b>	.0025	.0000	.0001	*	*	*
<b>F test</b>	.0000	.0000	.0000	*	*	*
<b>KS Test</b>	.5948	.0000	.0038	1.000	1.000	1.000

NOTE: \*One of the two compared data sets had no variance, so P-value could not be computed.

**Table 25. Missile Miss Distance Correlation Matrix**

Test	System 1			System 2			System 3		
	P2-HITL	P2-OAR	HITL-OAR	P2-HITL	P2-OAR	HITL-OAR	P2-HITL	P2-OAR	HITL-OAR
<b>North Dry</b>									
T test	.1146	.0448	.0168	.0002	.0000	.9392	.0099	.1904	.1592
F test	.0000	.0000	.0000	.0000	.0000	.0029	.0781	.0000	.0000
KS Test	.4789	.0000	.0000	.0002	.0000	.0229	.1746	.0000	.0000
<b>South Dry</b>									
T test	.9566	.1270	.1262	.0214	.0000	.1779	.0786	.0000	.0000
F test	.0316	.0000	.0000	.0000	.0000	.1381	.0609	.0001	.0105
KS Test	.4860	.8392	.0324	.0000	.0000	.0000	.2228	.0000	.0156
<b>North Wet</b>									
T test	.0985	.0000	.0002	.0210	.0499	.1373	.4125	.9685	.0000
F test	.1416	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
KS Test	.0856	.0000	.0082	.0000	.0000	.0020	.0000	.0000	.0000
<b>South Wet</b>									
T test	.0000	.0000	.0021	.0039	.0206	.1596	.6817	.1014	.0064
F test	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
KS Test	.0000	.0000	.0000	.0000	.0001	.0000	.0000	.0000	.0000

NOTE: \*One of the two compared data sets had no variance, so P-value could not be computed.

### 6.3.1 Conclusions

Close correlation between data sets from different test phases distinguished by P-values greater than the tester's (or reader's) level of risk suggests that the same EW Test performance measure population data can be studied successfully using the test techniques employed in either phase. Lack of correlation indicates that there was some unaccounted variable present in the test process in one or both phases that impacts that particular EW Test performance measure. Further study of other potentially impacting variables points to strong differences in data sets because of operator variance and differing threat representations between facilities. JADS could find no evidence in this analysis that geographical distribution measurably affected the results.

## 6.4 ADS Effects on EW Test MOP Summary

Table 26 summarizes the effects of ADS on the ten different MOPs. The table covers the general effects of data latency, data loss, data corruption, and operator variance. The last column also discusses methods used to circumvent the problems encountered with an ADS test for this MOP. For a more detailed explanation of the ADS effects on each MOP, please refer to the classified EW Test results report.

**Table 26. Effects of ADS on EW Test Measures of Performance**

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
<b>Correct Threat ID</b>	No JADS effect because threat ID is based on pod response to threat without respect to time.	If data packet is lost, ID may be missed.	If threat ID is corrupted in packet, data may be lost or false.	No impact.	Send threat ID messages reliable, and obtain text files of digibus monitor data.
<b>Correct Threat ID Response Time</b>	Delays in mode changes will affect apparent response time in real hardware.	If ID message is lost, no response time will be given.	If header time is corrupted, response time value will be false.	If not slaved to target at beginning of engagement, response times may be erroneous. (Potential exists to calculate response time based on received power that can be affected by the tracking error. Potential ADS problems discussed under tracking error.)	Use special instrumentation to remove latency from data samples. Use header time to correct for latency in DSM applications.
<b>Correct ECM Technique Selection</b>	No JADS effect because ECM technique is based on pod response to threat without respect to time.	If data packet is lost, ID may be missed.	If ECM ID is corrupted in packet, data may be missed or false.	No impact.	Send ECM ID messages reliable, and obtain text files of digibus monitor data.

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
<b>Correct ECM Technique Selection</b>	Delay in mode changes will affect apparent response time in real hardware.	If ECM message is lost, no response time will be given.	If header time is corrupted, response time value will be false.	If not slaved to target at beginning of engagement, response times may be erroneous.	Use special instrumentation to remove latency from data samples. Use header time to correct for latency in DSM applications.
<b>Jamming-to-Signal Ratio</b>	No JADS impact because J/S values are based on position to threat only at AFEWES. (Potential exists for errors to be introduced if AFEWES threat actions are combined with aircraft position in a different facility to determine the result. This potential exists for "measured" data derived from antenna patterns, transmitted powers, pointing angles, and platform positions where the positions are not in the same facility/frame of reference.)	If many samples are lost, J/S curve will look poor. (For measured J/S, insight into the quality of other measures is compromised.)	If many samples are corrupted, curve will look poor. (For measured J/S, insight into the quality of other measures is compromised.)	Since values are calculated at AFEWES regardless of tracking error, no JADS impact. For measured J/S abnormally poor tracking will produce abnormally low J/S.	Use real-time analysis methods to watch data as they arrive and find anomalies. Ensure aircraft and threats are in same reference frame.

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
RMS Tracking Error	No JADS impact by latent data because dead reckoning algorithm at AFEWES negates most latency effects. (Potential exists for errors to be introduced if threat pointing angles are combined with aircraft position in a different facility to determine the result. Spike can be introduced into data as dead reckoning position is replaced by "actual" position.)	No JADS impact by lost data because dead reckoning algorithm at AFEWES negates most data loss effects. (Potential exists for errors to be introduced if threat pointing angles are combined with aircraft position in a different facility to determine the result. Spike can be introduced into data as dead reckoning position is replaced by "actual" position.)	Potential exists that corruption of samples will invoke the dead reckoning algorithm or become spikes in tracking error as aircraft is moved to the incorrect position. Potential exists that data loss could prevent recovery of tracking error where threat pointing angles are combined with aircraft position in a different facility.	Operators have significant variance in manual modes. Abnormally poor tracking can skew results.	Use as much computer control (AUTO mode) as possible. Dead reckoning algorithms assist in minimizing latency and loss effects. Use real-time analysis methods to determine anomalies in data. Ensure aircraft and threats are in same reference frame.

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
<b>Number of Breaklocks (B/L)</b>	No JADS impact because breaklocks are based on mode changes, not time of mode changes.  (Potential exists for errors to be introduced if breaklocks are determined by exceeding tracking error thresholds and tracking error is derived from threat pointing angles which are combined with aircraft position in a different facility.  Potential exists for breaklocks to be induced as dead reckoning position is replaced by “actual” position.)	If mode changes are lost, data samples may be inaccurate.  (Potential exists for errors to be introduced if breaklocks are determined by exceeding tracking error thresholds and tracking error is derived from threat pointing angles which are combined with aircraft position in a different facility.  Potential exists for breaklocks to be induced as dead reckoning position is replaced by “actual” position.)	If mode change information is corrupted, data samples may be false. Corrupted tracking error may affect this measure if breaklocks are determined by exceeding tracking error thresholds and tracking error is derived from threat pointing angles which are combined with aircraft position in a different facility.  Potential exists for breaklocks to be induced as dead reckoning position is replaced by “actual” position.)	If operator changes between auto and manual modes inconsistently, MOP will be impacted.  Abnormally poor tracking will affect this measure if breaklocks are determined by exceeding tracking error thresholds.	Use reliable transmission of mode changes, and train operators so reactions and operations are consistent.  Ensure aircraft and threats are in same reference frame.
<b>Reduction in Engagement Time</b>	No JADS impact because engagement time is based on mode and tracking error.  (Potential exists for tracking error and breaklock problems discussed above to affect the result of this measure.)	Not directly impacted by data loss, but effects on tracking error can cause engagement time to be affected.	Unless tracking error and mode changes are severely impacted, corruption will have little to no impact.	Consistent operator action is key. Variance will severely impact this MOP.  Abnormally poor tracking or increases in breaklocks will skew results.	Ensure operator training to minimize fluctuations in operator reactions.  Ensure aircraft and threats are in same reference frame.

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
<b>Reduction in Missiles Launched</b>	No JADS impact because MOP is based solely on missiles fired.	Lost missile performance message will affect MOP. If TSPI drops for extended period causing B/L and tracking error to be affected, this MOP will be affected as well.	No impact unless corruption in other data causes B/L or large tracking error values.	Operator actions are key to this MOP. If firing patterns are done inconsistently, MOP will be affected. Abnormally poor tracking or increases in breaklocks will reduce valid shot opportunities.	Ensure consistent operator actions.
<b>Missile Miss Distance</b>	No JADS impact since missile and aircraft are in same reference at AFEWES. (Potential exists to combine missile flight path in one reference to aircraft in another reference. Also may be affected when tracking error is affected by latency as discussed above. Finally, changes in jamming onset may alter effectiveness of some techniques against some systems.)	Unless missile performance message is lost, MOP can only be impacted by tracking error in JADS. (Potential exists to combine missile flight path in one reference to aircraft in another reference. Also may be affected when tracking error is affected by data loss as discussed above. Finally, changes in jamming onset may alter effectiveness of some techniques against some systems.)	Corruption of missile performance messages can cause MOP to be corrupted in JADS. Likewise, corruption of tracking error or aircraft position can affect MOP when missile flight path in one reference to aircraft in another reference are combined to produce the MOP.	Missile launch during bad track can cause large variance in data samples. Aborting missiles mid-flight and not reporting them also causes large variance in data samples.	Ensure consistent operator actions. Use common reference frame for aircraft and missile.



## 7.0 Lessons Learned

### 7.1 Execution Phase: Pretest

#### 7.1.1. Software Acceptance Testing

**Problem Statement.** Software acceptance testing of ADS components in their stand-alone mode did not uncover problems once they were integrated into the ADS environment.

**Impact to Phase 2.** Software acceptance testing was not planned as part of the software development effort. Formal testing was thought to be too costly and too late in development to be effective. JADS planned on using in-process reviews with each developer to gain enough insight and cross communication to get the right software products developed. However, when JADS was unable to gain insight into the software development and received obvious indications that there were flaws in some of the software items, JADS elected to use acceptance testing. Because of cost and schedule constraints, the scope of these tests was limited to the development environments and to the test sets that were available at the time. These acceptance tests did not address all software requirements. For example, the acceptance test did not consider the operational modes of the jammer digital system model as executed in the ADS environment. The acceptance testing also did not stress the model to the level of execution encountered within the ADS environment. This resulted in a model that functioned well in stand-alone mode, but was marginal when integrated into the ADS environment and operated according to the test procedures.

Acceptance testing did provide a more solid basis for verification and validation efforts. The limited acceptance test did address several key requirements such as correct calculation of received power and correct calibration. The accreditation board had the results of the acceptance testing to better understand JADS' software needs.

Finally, acceptance testing allowed a convenient point for establishing configuration baselines and for transferring control of those baselines to JADS.

**Planned Corrective Action for Phase 3.** Acceptance testing will be better planned in Phase 3 even though we still have limited test cases and tools. The new software will be acceptance tested as part of the V&V plan. Formal baselines will be established after completion of the acceptance tests.

**Implication to Future ADS-Based Tests.** Acceptance testing of federate software is a recommended practice. These acceptance tests should be designed to test the software in its intended mode of operation, and to test all requirements of the software. Testing can encourage the developer to fix problems before they impact the test, it provides an excellent mechanism for supporting the V&V of the federation by proving the federates are built correctly and satisfy the

needed simulation requirements, and acceptance tests provide a clear event to which configuration management milestones can be tied.

### **7.1.2 Abbreviated Statements of Work (SOW) and Distributed Simulation Testing**

**Problem Statement.** Abbreviated statements of work and reduced deliverables resulted in differences in expectations between contractors and the government.

**Impact to Phase 2.** The loosely defined statement of work allowed the analysis team to continue to refine requirements for a critical piece of software past when it needed to be finalized. Several measures of performance needed to be calculated in a nontraditional fashion to measure ADS effects. This proved more difficult than expected. Since delivery schedules were not clearly defined, the contractor permitted these discussions to go on well beyond the time needed to code and test the software to meet the government's expected delivery date. All parties were trying to get the best insight into ADS effects on the EW measures while balancing impacts to the software. The problem was resolved when the government program manager froze software requirements and provided the contractor a specific delivery date. A second impact was related to the level of on-site test support. The loosely defined statement of work allowed the contractor to reallocate on-site resources earlier in the test design to support other test activities. The reallocation was discussed with the government, however the impact to on-site support during Phase 2 was not explicitly negotiated. As a result, the government received less support than expected.

**Planned Corrective Action for Phase 3.** All software will be developed and delivered prior to Phase 3.

**Implication to Future ADS-Based Tests.** Frequently the government will know only in general terms what is needed to execute tests in a geographically distributed environment. Test design has to mature to identify the specific capabilities that each facility will provide before specifications can be created. This generally precludes creating good performance specifications prior to contract award. Sufficient tasking must be included in the SOW to ensure that government interests are covered and the contractor lead has a leverage tool to use on management to ensure the work is executed on time with good quality. Sequential contract awards may be used to mitigate risks associated with loose statement of works.

### **7.1.3 Maintaining a Schedule for an Advanced Distributed Simulation Test Execution**

**Problem Statement:** EW tests typically require several critical assets. Delays in one asset can impact the overall test schedule. This becomes a larger problem with ADS, since delays require rescheduling multiple facilities, each with their own time and asset constraints.

**Impact to Phase 2.** The Phase 2 test schedule slipped because of delays in obtaining data from the first phase of the JADS EW Test. The jammer digital system model required response time data from the first test phase for calibration. The first two attempts to collect these data at the open air range and then during the hardware-in-the-loop test failed and required a third test event

at a systems integration laboratory. These data however were collected much later than required to prepare for Phase 2 test execution. As a result, this test phase was delayed.

**Planned Corrective Action for Phase 3.** We will aggressively manage the schedule and work with our supporting organizations to ensure that resources are ready and in place to support the test as scheduled. We will demonstrate that the test is nonexecutable if it slips. No organization wants to be responsible for canceling the test event.

**Implication to Future ADS-Based Tests:** Schedule may be the hardest factor in ADS testing to control because it is influenced by internal factors (e.g., the ability of the different facilities to work together to identify and solve problems) as well as external factors (e.g., other tests that the facility has to support and how much influence those tests have). Aggressive management of all development efforts and deliverables, effective risk management, and starting the effort with enough cost, schedule, and performance trade space are all essential ingredients to success.

#### 7.1.4 Software Quality Assurance Reality

**Problem Statement.** Well-defined quality software practices are important for any software development, however when working with multiple facilities as with an ADS test, strict adherence to practices is necessary to ensure success. In addition, processes for assessing software quality (e.g., independent acceptance test) are needed to ensure that each ADS component operates as expected.

**Impact to Phase 2.** No plan was in place to ensure software quality. JADS relied on each developer's internal practices to produce quality software. JADS attempted to gain insight into software development at each facility but failed. (See 7.1.1.) Post-development quality measures had to be implemented to inspect delivered software. Several problems were identified with the DSM that should have been identified early in the software development process. Specifically, cases were found where developers misinterpreted software requirement specifications and the ICD. These problems could have been found by closer monitoring of the software development process particularly in the area of requirements management.

**Planned Corrective Action for Phase 3.** We will get more involved in the software development of the remaining federate. Better daily contact can prevent errors from going undetected until the actual test event.

**Implication to Future ADS-Based Tests.** Stricter contractual requirements may be needed for organizations that are known to use ad hoc development processes. Critical software should be developed by companies with proven subject matter experience and sound software development practices.

### **7.1.5 Strong Systems Engineering Function in ADS-Based Test Design**

**Problem Statement.** Lack of a single independent systems engineer during the development of Phase 2 software design and integration resulted in unnecessary confusion.

**Impact to Phase 2 Test.** JADS had assumed the lead systems engineering role throughout the test. During Phase 2 execution the responsibility of system engineering unofficially transferred to other IPT members. Quite often, IPT members were also responsible for performing tasks to begin development and delivery of several key software elements. This placed them in awkward positions for remaining unbiased and independent during integration. The systems engineer had to be free to identify and aggressively solve problems. This was best done by using an independent systems engineer.

**Planned Corrective Action for Phase 3.** JADS will reassume the role of systems engineer in Phase 3.

**Implication to Future ADS-Based Tests:** ADS requires strong systems integration and systems engineering. This should be kept out of the hands of anyone supplying items to be integrated. If the sponsor is unable to provide the expertise, outside engineering should be obtained. Subject matter experience and knowledge of computers and communications technology are essential for the systems integrator.

### **7.1.6 Reliable Distributor Servicing Multiple Federates**

**Problem Statement.** When JADS began working with the RTI, complete documentation on the correct use of all the RTI services and calls was not available. JADS was surprised to learn post-test that the reliable distributor servicing the federates located in Albuquerque was incorrectly implemented. The following is a detailed discussion of the reliable distributor and how JADS implemented it for the federates in Albuquerque.

Normally, every federate includes a reliable distributor (reldistr) based on the Internet TCP, since the RTI best effort communications mechanism provides neither guaranteed delivery to all message recipients nor in-order message delivery. The reldistr is used to send reliable data, i.e., guaranteed, in-order delivery from one federate to one or more other federates.

During the analysis of Phase 2 data loss and data delay events, there were many instances of differential latencies for reliable messages sent from a federate on one test node to two or more federates on the other nodes. For example, a latency-sensitive jammer technique command message sent by the DSM federate at ACETEF might arrive with a normal latency at AFEWES and two of the JADS federates but be delayed to the other two JADS federates by hundreds of milliseconds or even seconds. When DMSO technical support was queried about such anomalies, they advised JADS that the Phase 2 test actually had three reldistrs running on the RFENV host at the JADS node in addition to single reldistrs in the federates at the AFEWES and ACETEF nodes.

In an effort to minimize the amount of traffic on the WAN, the DMSO liaison for JADS recommended that a single reliable distributor for the federates in the TCAC be used during Phase 3. This also was desirable to eliminate some types of differential latency problems. The RFENV federate was chosen to host the reliable distributor for the TCAC.

**Impact to Phase 2.** The RFENV federate had to be started first, since all other federates would attempt to connect to its reldistr. Due to the two redundant reldistrs in the RTIEXEC and FEDEX on the same SGI O2 host, redundant TCP connections were apparently created (based on post-test network packet sniffer evidence) between the reldistrs on RFENV and those at AFEWES and ACETEF. The extra reldistrs and the redundant network pathways probably were the cause of some differential latency events during Phase 2.

**Planned Corrective Action for Phase 3.** The RTIEXEC has its own reldistr, so for Phase 3 all federates in the TCAC will be configured to use the RTIEXEC reldistr. However, due to a problem with RTI Version 1.3 Release 5, this will require that the RTIEXEC be started with one version of the RTI.rid file, which then has to be replaced by a second version before the FEDEX and the RFENV federate are started. This minor inconvenience will be handled by means of a UNIX shell script.

**Implication to Future ADS-Based Tests.** There are two primary implications to ADS-based tests. First, federations with multiple federates on a LAN should consider using a single reldistr per LAN. Second, RTI developers need to clearly document how to correctly implement nondefault configurations so that federations can take full advantage of the RTI features. Further implications are discussed below.

Designers, instrumenters, and executors of real-time, performance federations with latency-sensitive messages sent via the RTI reliable communications protocol to two or more federates on other distributed test nodes need to carefully consider the potential consequences of differential latencies. That is because differential latencies can cause the federates to have different perceptions of if and when critical events happened.

The original RTI developer's decision to use TCP for reliable traffic may have unavoidable, long-term, negative consequences that may cause trouble for some real-time, performance-oriented HLA-based simulations. For example, during RTI performance testing leading up to Phase 2, JADS learned that TCP implementations differ significantly, not only among those of different vendors, but also among different operating system version releases from the same vendor. A significant example of this is in the availability of the so-called TCP\_NODELAY option that would allow the reldistr's TCP to acknowledge incoming TCP segments without delay. This option was not available in SGI's IRIX 6.3 operating system but is available in IRIX 6.5 Sun Solaris and some other operating systems. Use of this option within the RTI and by the federate developers for non-HLA federate components (e.g., the DSM PC software) probably would have reduced the latencies of reliable messages.

Also, it is not at all clear that RTI developers using TCP for reliable distributor implementations have any means to guarantee that the TCP underlying a transmitting reldistr sends all copies of a reliable message intended for two or more recipients with minimal delay between outbound copy over a separate TCP connection. That is because the TCP protocol was never developed with this type of performance requirement in mind. It is also unclear as to whether intermediate reldistrs might introduce additional differential latencies because of a lack of control over the details of TCP actions on two or more independent TCP connections (a TCP connection consists of two pairs of IP addresses and port numbers, one for the source and one for the destination).

### 7.1.7 RTI Reliable Traffic

**Problem Statement.** Federation performance varied as the mix of reliable and best effort data changed. Through trial and error fewer problems with latency and data loss were noted if less reliable traffic was being published within the federation. However, this was subjective as there were no tools to test the performance envelope of the architecture.

**Impact to Phase 2.** Link health check messages were changed to be published best effort in an attempt to improve performance. This change was made late in the integration effort to further tune the architecture with the real federates.

**Planned Corrective Action for Phase 3.** Link health check messages will be published best effort for Phase 3 also.

**Implication to Future ADS-Based Tests.** Federations should experiment with different transport modes to determine the optimum mix of transport modes. RTI developers should have tools or performance measurements to guide federation developers as they design and integrate their architectures.

## 7.2 Execution Phase: Pretest, Test, and Post-Test

### 7.2.1 Conformance to a Well-Written ICD Is Necessary to Complete an ADS Exercise.

**Problem Statement.** An ICD was developed for the JADS federation to guide software developers. Two problems were identified related to this ICD: nonconformance to the ICD and differences in interpretation of complex concepts.

#### Impact to Phase 2 Test

##### Pretest Impact

The description of the coordinate transformation was agreed to be acceptable by all participants, however, when implemented, facilities developed different interpretations. The problem was finally resolved when JADS provided sample transformation pairs for testing each facility's

algorithm. These sample data points should have been included in the JADS Federation ICD to avoid confusion.

#### Test and Post-Test Impact

In some instances software was developed that did not conform to the ICD. Due to the lack of detailed acceptance testing (see 7.1.1) these nonconformance problems were not found until very late in the integration process. As a result, a decision had to be made as to whether to bring the software into conformance or to change the ICD in order to maintain the test schedule. For example, problems with the federate message sequence numbers illustrate both the test and post-test impacts. For each instance of a simulation object, the federates should have used, in outgoing messages, sequence numbers starting from 1 and incremented by 1 for each successive message. However, because of a combination of ambiguous ICD wording and lack of early ICD compliance testing and enforcement, the terminal threat hand-off and threat federates transmitted message sequences that did not conform to the same sequence numbering scheme.

During test execution this became a problem with the DSM PC's real-time error checking for incoming source mode change (SMC) messages. The DSM used the sequence number to detect missing and out-of-order messages. Since these sequence numbers were not set correctly, the error reports were misleading and therefore ineffective.

During the post-test analysis, not having proper message sequence numbers for several message types made it more difficult to detect and analyze runs with data loss and data delay problems for the ADS analysis. In particular, it greatly complicated the calculation of overall latencies for the critical combination of outgoing SMC messages and the corresponding jammer technique command messages generated by the DSM.

**Planned Corrective Action for Phase 3.** Message sequence counters will be corrected for both the terminal threat hand-off and threat federates. The ICD will changed to be less ambiguous.

**Implication to Future ADS-Based Tests.** Perhaps the most important lesson learned from Phase 2 was the critical importance of careful planning and preparations at the earliest stages of the program. (This, of course, is a well-known lesson learned for almost all software development programs.) It is better to avoid problems, since there may not be enough time and/or money to find and fix them later. This seems especially true for ADS programs. The nature of ADS brings multiple facilities together, each having their own development style and practices, and each bringing a potentially different understanding of the problem. (This is very similar to having multiple facilities working together to develop a single software package.) Any actions that reduce ambiguity in the interface design will reduce the risk to the program. This is very important for ADS-based tests since it may be difficult to slip test schedules when multiple facilities are involved. Having a good ICD and enforcing compliance with it from the start is very important.

## 7.3 Execution Phase: Test

### 7.3.1 Time Synchronization

**Problem Statement.** ADS was not able to completely solve time synchronization issues in the federate computers using time cards. In theory, the hardware cards should provide the most accurate time synchronization available. In practice, some implementations proved more robust than others and verifying time synchronization across a wide area network proved to be elusive. These issues are discussed in detail below.

The most effective configuration of the BanComm cards was not implemented for time synchronization on either the UNIX-based or the PC-based hosts. In addition, there were problems with BanComm's hardware, BanComm's software, and with one JADS contractor's attempts to write software to use the BanComm cards.

The software executing on the SGI O2s read time directly off the BanComm cards via the JADS contractor-developed driver software. This provided the most accurate time synchronization solution. However, the method used to obtain time information (via overloading of an IRIX operating system call) had the limitation that it did not provide any means for the federates to query the BanComm card as to whether it was actually using the IRIG B time code input signal (the desired state) or free running using its internal crystal oscillator.

However, on the PCs, BanComm-provided software was used to synchronize PC system time to BanComm card time. This was not very accurate, and in some cases, time on the PCs was off by as much as 60 ms. In addition, this software did not synchronize the system time immediately when Windows 95 or Windows 98 was started or restarted, which apparently caused several aborted runs because of the time on an ADRS PC being unsynchronized. Also, for the PCs, there was still the problem of determining when the BanComm lost its signal and was free-running on its internal oscillator.

Finally, JADS lacked an adequate method of detecting time synchronization problems in real-time during federate execution runs. Only in cases where severe symptoms were produced by time synchronization, problems such as bursts of platform federate live entity state and threat performance messages caused by a start time in the past (for platform) were noticed immediately and corrected.

**Impact to Phase 2.** Data that were time stamped on the PCs (ADRS and DSM) were only judged "good enough." There was a lot of variation in the time value that originated on the PCs. This did not impact the ADRS PCs as they only used the time stamp in the start command, telling all federates to start at some time in the future. However, the DSM PC did exhibit some odd behavior that affected calculation of jammer response times.

**Planned Corrective Action for Phase 3.** There is no corrective action to be taken.

**Implication to Future ADS-Based Tests.** If you are going to use hardware for time synchronization (e.g., BanComm cards) obtain time directly from the card. You may have to write device drivers to get this capability. You also need to resolve how you will measure time synchronization differences. However, other alternatives exist. The Network Time Protocol (NTP) software (xntp for UNIX hosts; NTP time for PC hosts) provided an easy to use method to synchronize system clocks to a time source. In other JADS tests the system clock could be kept within 1 ms of the time source. This software does require time and attention to reach this level of performance. However, it keeps statistics on how well it is keeping time and it's free.

## 7.4 Execution Phase: Post-Test

### 7.4.1 RTI Best-Effort IP Multicast Groups

**Problem Statement.** Details on how the RTI handles its communications are deliberately withheld from the user. This is done to encourage users to treat the RTI as a black box and adhere to the interface specification. This works for most users, however, T&E has a need to know how communications are handled. JADS was surprised to learn post-test how the RTI really created multicast groups. Instead of separate multicast groups being established according to actual publish/subscribe topology, JADS best effort data were sent in a single multicast group to which all federates were connected. Each local instance of the RTI had to deal with all messages even if its federate did not subscribe to all messages. This should have been known in early design so that different implementations could have been tested. The following is a detailed discussion on how this worked within RTI 1.3 release 4 and 5. Also, there is a discussion of the data losses that were apparent and how the multicast implementation may have contributed.

When the RTI executive (RTIEXEC) starts execution, it transmits Internet Group Management Protocol (IGMP) “report” messages to join several IP multicast groups which, for the JADS federation, had Class D Internet addresses of the form 224.253.xxx.yyy. The federation executive (FEDEX) does the same when it begins, and so does each federate as it joins the federation. These IP multicast groups provide, via the UDP, the RTI’s one-to-one and one-to-many best effort communications infrastructure.

The RTI within each federate uses the stream map in the RID file (i.e., the file RTI.rid) to determine to which multicast group a particular type of best effort message should be sent to reach a specific federate or group of federates. The specific multicast groups joined by a federate depend on when it joins versus the other federates. Also, as new federates join (or joined federates resign), the RTI dynamically redirects best effort traffic within the established multicast groups.

After Phase 2, JADS discovered this behavior by using network packet sniffers on the SGI O2 hosts and eventually learned from the RTI developer that the stream map in the RID file provided to JADS caused all federates joining after the third federate to stop joining new multicast groups in addition to those already created. Instead, they joined a broadcast multicast group (224.253.1.0), and federation traffic formerly sent to specific multicast groups was redirected to that group. The result was that all federates received, even if they did not subscribe, almost all

best effort messages, and the local RTI component (LRC) within the federates had to process and discard those unwanted messages. Thus, for example, the LRC in the hand-off federate, which does not subscribe to threat performance, had to receive, process, and discard five 20-hertz message streams from the platform and AFEWES federates.

During Phase 2, there were many instances of best effort data losses that were unusual in two ways: they were one-way losses, meaning that messages between two or more federates were lost in one direction, but not in the opposite direction; and they were selective losses, for example, the DSM federate did not receive link health, live entity state, and threat performance messages from the platform federate at JADS but did receive link health messages from the other three JADS federates.

These losses cannot be explained by network problems such as a short outage on one of the T-1 lines, loss of crypto synchronization, etc., since those problems would affect all best effort traffic in both directions between two test nodes. This suggested that these selective, one-way best effort data losses might have been due to some problem with the RTI's use of IP multicast groups. Or, they might have been caused by "pruning" of some IP multicast addresses by the protocol independent multicast-dense mode (PIM-DM) routing protocol that the JADS routers used.

**Impact to Phase 2.** Due to lack of adequate documentation for the RTI RID file, JADS unknowingly used a RID file with a stream map that was probably not appropriate for a federation with six or seven federates. As a result, almost all our best effort data were sent to all federates, unnecessarily loading some of them.

Perhaps due to IP multicast-related bugs in RTI Version 1.3 Release 4, and/or router protocol pruning of RTI IP multicast addresses, JADS experienced many unusual, selective, one-way best effort data loss events. For runs 36 and 107, these events had consequences that caused unacceptable response times for some DSM jammer technique commands.

**Planned Corrective Action for Phase 3.** The RTI.rid file could be modified with a new stream map to provide more multicast groups to the federation. The RTI developer's suggestion of using data distribution management could have been accepted but was rejected for Phase 3 for the same reason that it was not used for Phase 2: namely, there appeared to be a significant risk of adding unacceptable latencies to the real-time, performance-oriented federation.

Regarding the second problem, PC-based network packet sniffers on the LANs leading to the routers were added at all three nodes to improve the instrumentation.

**Implication to Future ADS-Based Tests.** Just because federation designers are careful about which federates subscribe to data (in an effort to reduce WAN traffic), doesn't mean that the data aren't being sent to the federate anyway. Federation designers need to think carefully about the instrumentation for monitoring their federations, and that instrumentation should be in place well before the start of formal integration testing. Phase 2 showed that RTI loggers, DIS-style passive loggers, Internet ping probing, and network error printouts provide, at best, only circumstantial

and limited evidence to diagnose the root causes of most data latency and data loss problems for HLA-based distributed simulations.

RTI developers need to document how the RTI establishes multicast groups so that federation designers can take full advantage of what the RTI has to offer. High performance federations can't treat the RTI as a black box.



## **8.0 Conclusions/Recommendations**

This report describes the ADS implementation, development and integration process, ADS and correlation results, and lessons learned for Phase 2. Since this is the first of two ADS-based tests, conclusions to the JADS issues are not presented. These will be fully addressed in the Phase 3 report. While correlation results are included, the underlying EW Test data are not presented. These results are classified and will be contained in a separate report documenting both Phase 2 and Phase 3.

Phase 2 used an HLA-compliant ADS architecture to successfully recreate both an open air test and a hardware-in-the-loop test. The Phase 2 architecture successfully integrated a digital system model representing an early representation of a self-protection jammer with the high fidelity threats at AFEWES. This implied that ADS might be used to address the EW test process issues of correlation and fidelity. This was not enough to fully discuss the utility of ADS for EW. Complete discussion on the utility of ADS to EW testing will be the subject of the JADS EW Test final report.

Examination of the MOP data indicated that there were isolated incidents of ADS impacting the test results. Aircraft position data dropouts seen in integration forced JADS to add a simple dead reckoning algorithm into the AFEWES gateway. However, when the data resumed, the aircraft was immediately moved to the updated location. In one instance, this caused a problem in the flyout model of a missile. The interviews indicated that there were some odd aircraft behaviors at the start of several of the scripts (but outside of the core engagement area) that operators deemed unrealistic. These did not affect the EW Test MOPs. The data and interviews indicated that there were no consistent ADS-induced biases or flaws that would make the data invalid.

These results were expected since the test design took advantage of the unique capabilities of the AFEWES facility. Properly designed ADS architectures should not impact test results. All measures except for the jammer timing measures were taken within the AFEWES facility which avoided many of the pitfalls that could affect ADS-based tests. The jammer timing measures were measured across facilities. Combining events in separate facilities in a single measure impacted measures where the transmission latency was significant in comparison with the duration of the event in a non-ADS environment. However, the JADS test team was unable to assess the impact of this because of shortfalls in the DSM design. This will be addressed in the Phase 3 report. Data latency in excess of the design goal and lengthy bursts of lost aircraft position data did not affect the EW Test MOPs in any consistent, measurable fashion. This performance and the lack of impact were somewhat surprising. It pointed out how powerful and necessary dead reckoning for fixed rate data streams really was. The JADS architecture was an example of a well-designed ADS-based test.

There were limitations within the ADS architecture JADS created. Different jammer techniques and more reactive players (open rules of engagement for the threats and maneuvering aircraft) will require that the bursts of lost aircraft position data be resolved and latency performance be improved over what was observed in Phase 2. Predictive jammer techniques will also require

more of the jammer processing logic be collocated with the JETS at AFEWES. Several of the message structures and common data used in our test will have to be examined before being used in other tests. While all the message structures have room for growth, they need to be examined by future implementers to ensure the size and intent will meet the requirements of the new federation.

The most significant limitation to this architecture was the availability of high fidelity threats suitable for ADS-based testing. Low fidelity threats are not difficult to add to this architecture. However, to address the correlation and fidelity shortfalls of the EW test process, high fidelity threat representations are the key to the greatest benefit from this architecture. AFEWES used distributed simulation techniques within its facility to accomplish traditional testing. ADS simply allowed AFEWES to connect to other facilities or locations. The OAR used in Phase 1 had high fidelity threat simulators as well. However, these were not suitable in their current configuration to accomplish testing within the JADS architecture. RF injection into the threat for both target and jamming was key to this architecture. The second key was the infrastructure to tie the threats together to engage a common virtual target in a common synthetic environment. Neither of these was available on the OAR used in Phase 1. According to the CROSSBOW-sponsored Threat Simulator Linking Activity study, these types of high fidelity threat simulators are very scarce resources.

Phase 2 demonstrated that ADS tests create valid data when properly designed. ADS can be used to connect low fidelity, real-time digital system models with high fidelity threats. Closed-loop interactions that can tolerate 500 milliseconds or more of additional round-trip latency can be tested using the current HLA runtime infrastructure and commercial communications hardware. These features make ADS a useable tool for EW testers. However, the availability of suitable high fidelity simulators will ultimately determine how quickly ADS is integrated into the mainstream of EW testing.

## Appendix A

### Phase 2 Script Execution Matrix

When executing Phase 2, the matrices outlined in Section A1 were used when the SADS VIII and West X were manned at AFEWES. The profiles listed in Section A2 and A3 were used to test all four live threats. Section A4 was used when executing profiles with the SADS III and SADS VI M manned at AFEWES.

The mission and run designations were derived from the OAR mission profiles. The profiles are numbered *XY-ZZ*.

**X** designates which threats were manned at AFEWES (0 or 1 = all live threats, 2 or 3 = SADS III and SADS VI M live, 4 or 5 = SADS VIII and West X live).

**Y** designates the OAR mission used to generate the profile (5-9 = OAR missions 5-9, 0-1 = OAR mission 10-11).

**ZZ** designation is the number of the specific run from the OAR mission (1-20 = OAR runs 1-20). For example, profile 45-5 means a live SADS VIII and West X from OAR mission 5, run 5.

Profiles deviating from this scheme were most of the 14 karat runs, which were arbitrarily named.

#### **A1. SADS VIII and West X Live at AFEWES.**

The profiles in this section had the activation messages in the TTH for the SADS VIII and West X. The RFENV contained the mode messages to load the DSM for the SADS III and SADS VI M threat systems, which should not be manned.

**Table A1. Live SADS VIII and West X - Day 1 and 2**

NORTH				SOUTH			
Dry	Wet			Dry	Wet		
45-6	45-10	45-12	46-2	45-3	45-5	45-7	45-9
46-4	46-6	46-8	46-10	46-7	46-9	46-11	46-17
46-18	47-4	47-6	47-8	47-1	47-3	47-5	47-7
47-12	47-14	47-16	47-18	47-9	47-13	47-17	47-19
47-20	48-2	48-4	48-6	47-21	48-3	48-5	48-7
48-8	49-5	50-7	50-9	49-4	50-4	50-6	50-12
50-13	50-15	51-2	51-4	50-14	50-18	51-1	51-3
51-6	51-8	51-10		51-5	51-7	51-9	

**Table A2. Live SADS VIII and West X - Day 3 and 4**

NORTH				SOUTH			
Dry	Wet			Dry	Wet		
46-2	45-10	45-6	45-12	45-7	45-5	45-3	45-9
46-10	46-6	46-4	46-8	46-11	46-9	46-7	46-17
47-8	47-4	46-18	47-6	47-5	47-3	47-1	47-7
47-18	47-14	47-12	47-16	47-17	47-13	47-9	47-19
48-6	48-2	47-20	48-4	48-5	48-3	47-21	48-7
50-9	49-5	48-8	50-7	40-6	50-4	49-4	50-12
51-4	50-15	50-13	51-2	51-1	50-18	50-14	51-3
	51-8	51-6	51-10	51-9	51-7	51-5	

## A2. Fourteen Excursion Runs

If time is available, the following profiles can be executed to test the ability of ADS to handle a more erratic scenario. The description of each profile follows the mission and profile number.

**Table A3. 14 Karat Run Matrix - Day 5**

Mission -Profile	DESCRIPTION			
	Speed (kts)	Altitude (x1000 msl)	N- S	Notes
11-1	360	9 K	N	standard ROE
11-1	360	9 K	N	standard ROE
81-1	360	9 K	N	sites come up IAW SCM - simultaneous missiles at overlap
81-1	360	9 K	N	sites come up IAW SCM - simultaneous missiles at overlap
82-1	550	9 K	N	standard ROE
82-1	550	9 K	N	simultaneous missiles
83-1	720	9 K	N	simultaneous site call-up - fire at will
83-1	720	9 K	N	simultaneous site call-up - simultaneous missiles
9-5	360	9 K	S	standard ROE - aircraft (A/C) ascent to 15K
9-5	360	9 K	S	standard ROE - A/C ascent to 15K
84-1	360	6.5 K	N	standard ROE
84-1	360	6.5 K	N	standard ROE
85-1	360	20 K	N	standard ROE
85-1	360	20 K	N	standard ROE

IAW = in accordance with  
SCM = site controller matrix

K = thousand

Kts = knots

### A3. All Live Threats at AFEWES

Execute profiles in this section when all threats are manned at AFEWES.

**Table A4. All Threats Manned at AFEWES - Day 5**

NORTH				SOUTH			
Dry	Wet			Dry	Wet		
5-6	5-10	5-12	6-2	5-3	5-5	5-7	5-9
6-4	6-6	6-8	6-10	6-7	6-9	6-11	6-17
6-18	7-4	7-6	7-8	7-1	7-3	7-5	7-7
7-12	7-14	7-16	7-18	7-9	7-13	7-17	7-19
7-20	8-2	8-4	8-6	7-21	8-3	8-5	8-7
8-8	9-5	10-7	10-9	9-4	10-4	10-6	10-12
10-13	10-15	11-2	11-4	10-14	10-18	11-1	11-3
11-6	11-8	11-10		11-5	11-7	11-9	

### A4. SADS III and SADS VI M Live at AFEWES

This section lists the profiles when the SADS III and SADS VI M are live and manned at AFEWES. The TTH scripts contain the activations and deactivations for the SADS III, and the RFENV scripts contain the modes for the SADS VIII and West X to load the DSM federate.

**Table A5. SADS III and SADS VI M Threats Manned at AFEWES - Day 6 and 7**

NORTH				SOUTH			
Dry	Wet			Dry	Wet		
25-6	25-10	25-12	26-2	25-3	25-5	25-7	25-9
26-4	26-6	26-8	26-10	26-7	26-9	26-11	26-17
26-18	27-4	27-6	27-8	27-1	27-3	27-5	27-7
27-12	27-14	27-16	27-18	27-9	27-13	27-17	27-19
27-20	28-2	28-4	28-6	27-21	28-3	28-5	28-7
28-8	29-5	30-7	30-9	29-4	30-4	30-6	30-12
30-13	30-15	31-2	31-4	30-14	30-18	31-1	31-3
31-6	31-8	31-10		31-5	31-7	31-9	

**Table A6. SADS III and SADS VI M Threats Manned at AFEWES - Day 8 and 9**

NORTH				SOUTH			
Dry	Wet			Dry	Wet		
25-12	25-10	25-6	26-2	25-7	25-5	25-3	25-9
26-8	26-6	26-4	26-10	26-11	26-9	26-7	26-17
27-6	27-4	26-18	27-8	27-5	27-3	27-1	27-7
27-16	27-14	27-12	27-18	27-17	27-13	27-9	27-19
28-4	28-2	27-20	28-6	28-5	28-3	27-21	28-7
30-7	29-5	28-8	30-9	30-6	30-4	29-4	30-12
31-2	30-15	30-13	31-4	31-1	30-18	30-14	31-3
31-10	31-8	31-6		31-9	31-7	31-5	

**Appendix B**  
**Site Controller Matrix**

Condition	Range (nmi from IP)	SADS III	SADS VI	SADS VIII	WEST X
North bound					
1	0.0	ON	OFF	OFF	OFF
2	4.5	"	ON	"	"
3	6.0	"	"	ON	"
4	8.6	"	"	"	ON
5	13.6	"	"	OFF	"
6	16.0	OFF	"	"	"
7	17.6	"	OFF	"	OFF
South bound					
1	1.5	OFF	OFF	ON	OFF
2	3.7	"	ON	"	"
3	6.0	ON	"	"	"
4	16.0	"	"	OFF	"
5	17.5	"	OFF	"	"
6	21.0	OFF	"	"	"

IP = initial point

nmi = nautical mile



## Appendix C

### Acronyms and Definitions

413 FLTS	413th Flight Test Squadron, Edwards AFB, California
A/C	aircraft
AAA	anti-aircraft artillery
AATC	Air National Guard Air Force Reserve Test Center, Tucson, Arizona
ACETEF	Air Combat Environment Test and Evaluation Facility, Patuxent River, Maryland; Navy facility
ADRS	Automated Data Reduction Software
ADS	advanced distributed simulation
AFB	Air Force Base
AFEWES	Air Force Electronic Warfare Evaluation Simulator, Fort Worth, Texas; Air Force managed with Lockheed Martin Corporation
ALQ-131	a mature self-protection jammer system; an electronic countermeasures system with reprogrammable processor developed by Georgia Tech Research Institute
AMES	Automatic Multiple Environment Simulator at Eglin AFB, Florida
AMI	alternate mark inversion
APA	analysis plan for assessment
API	application program interface
ASCII	American Standard Code for Information Interchange
AWC	Air Warfare Center at Nellis AFB, Nevada, and Eglin AFB, Florida
B/L	breaklock
B8ZS	binary eighth zero substitution
BERT	bit error rate test
C4ISR	command, control, communications, computers, intelligence, surveillance and reconnaissance
COTS	commercial-off-the-shelf
CRM	communications resource manager
CROSSBOW	Office of the Secretary of Defense committee under the Director, Test, Systems Engineering and Evaluation
CSU	channel service unit
DAT	digital audio tape
dB	decibel
DD, DT&E	Deputy Director, Developmental Test and Evaluation
DEC	Digital Equipment Corporation
DIS	distributed interactive simulation
DMAP	data management and analysis plan
DMSO	Defense Modeling and Simulation Organization, Alexandria, Virginia
DoD	Department of Defense
dry run	the system under test is off
DSM	digital system model

DSU	data service unit
DT&E	developmental test and evaluation
E&M	analog voice signaling standard
EAV	early access version
ECCM	electronic counter-countermeasures
ECM	electronic countermeasures
EMVI	emitter mode verification instrumentation
ENVGEN	environment generator
ESF	extended super frame
EW/EW Test	electronic warfare; JADS Electronic Warfare Test
FAT	federate acceptance test
FEDEX	federation executive
FEPW	federation execution planners workbook
FIT	federate integration test
FOM	federation object model
GEM	general effectiveness model
GPS	global positioning system
GTRI	Georgia Tech Research Institute, Atlanta, Georgia
HITL	hardware-in-the-loop (electronic warfare references)
HLA	high level architecture
HSNPL	hardware, software, and network problem log
HUD	heads-up display
Hz	hertz
I/F	interface
I/O	input/output
IADS	Integrated Air Defense System
IAW	in accordance with
ICD	interface control document
ID	identification
IDNX™	Integrated Digital Network Exchange
IGMP	Internet Group Management Protocol
INS	inertial navigation system
IP	internet protocol; initial point
IPT	integrated product team
IRIG	Inter-Range Instrumentation Group
IRIX	operating system for the Silicon Graphics, Inc.
ISTF	installed systems test facility
J/S	jamming-to-signal ratio
JADS	Joint Advanced Distributed Simulation, Albuquerque, New Mexico
JETS	JammEr Techniques Simulator
JMASS	Joint Modeling and Simulation System
JT&E	joint test and evaluation
JTF	Joint Test Force, Albuquerque, New Mexico
K	thousand
Kbps	kilobits per second

K-S	Kolmogorov-Smirnov test
kHz	kilohertz
KIV	AlliedSignal embedded KG-84 (a family of communications security equipment) communications security module
kts	knots
LAN	local area network
LES	live entity state
LHC	link health check
LRC	local runtime infrastructure component
Mb	megabit
MB	megabyte
Mbps	megabits per second
MHz	megahertz
MOE	measure of effectiveness
MOP	measure of performance
ms	millisecond
msl	mean sea level
MTI	moving target indicator
N&E	network and engineering
nmi	nautical mile
NRZ	nonreturn to zero
NSA	National Security Agency
NTP	network time protocol
OAR	open air range
OSD	Office of the Secretary of Defense
OT&E	operational test and evaluation
PC	personal computer
PCM	pulse code modulation
PIM-DM	protocol independent multicast-dense mode
PTP	program test plan
P-value	probability value
PX	packet exchange
QAVP	quad-analog voice processor
R/P	receiver processor
RAD	company that manufactures the voice signal converter
RCS	radar cross-section
reldistr	reliable distribution
RF	radio frequency
RFENV	radio frequency environment
RID	runtime infrastructure initialization data
RMS	resource management system
ROE	rules of engagement
RPSIM	radar processor simulation
RTC	reference test condition
RTI	runtime infrastructure

RTIEXEC	runtime infrastructure executive
SAC	senior advisory council
SADS	Simulated Air Defense System
SAM	surface-to-air missile
SCM	site controller matrix
SGI	Silicon Graphics, Inc.
SIL	system-in-the-loop; system integration laboratory
SMC	source mode change
SME	subject mater experts
SNMP	Simple Network Management Protocol
SOW	statement of work
SPAG	software-programmable antenna pattern generator
SPECTRUM®	a network analysis package developed by Cabletron Systems
SPJ	self-protection jammer
SRS	software requirements specification
STEP	simulation, test and evaluation process
SUT	system under test
T&E	test and evaluation
T/E	tracking error
T-1	digital carrier used to transmit a formatted digital signal at 1.544 megabits per second
TAB	technical advisory board
TAMS	Tactical Air Mission Simulator
TAP	test activity plan
TCAC	Test Control and Analysis Center, Albuquerque, New Mexico
TCF	test control federate
TCP	transmission control protocol
TMC	test management center
TP	threat performance
TRR	test readiness review
TSPI	time-space-position information
TTH	terminal threat hand-off federate
TTL	time-to-live
TTR	target tracking radar
UDP	user datagram protocol
UTC	universal time code
V&V	verification and validation
VSC	voice signal converter
WAN	wide area network
WEST	Weapon Evaluation Simulated Threat
wet run	the system under test is on
WTR	Western Test Range
Y2K	year 2000